HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

--------***--------

HÀ HẢI NAM

BÀI GIẢNG

PHÁT TRIỂN HỆ THỐNG THÔNG TIN QUẢN LÝ

HÀ NỘI 12-2014

LỜI NÓI ĐẦU

Các hệ thống thông tin quản lý vận hành trong mọi tổ chức, đơn vị nhằm cung cấp thông tin chính xác, kịp thời đúng nơi cho đúng đối tượng người dùng. Hệ thống thông tin quản lý hỗ trợ hiệu quả trình ra quyết định của các nhà quản lý

Mục đích của bài giảng này là giới thiệu tới sinh viên các các kiến thức, kỹ năng cơ bản về quản lý và công nghệ cần thiết để xây dựng các hệ thống thông tin quản lý. Những kiến thức nền tảng giới thiệu trong bài giảng này là kiến thức chuẩn bị cần thiết cho sinh viên tiếp cận với các chủ đề nâng cao trong thiết kế, xây dựng các hệ thống thông tin quản lý nâng cao.

Bài giảng gồm năm chương bao quát các nội dung kiến thức sau:

- Cấu trúc hóa dữ liệu với mô hình thực thể liên kế và UML
- Truy vấn dữ liệu
- Tổng hợp dữ liệu
- Hiển thị dữ liệu
- Hỗ trợ ra quyết định quản lý

Trong quá trình biên soạn tài liệu, mặc dù tác giả đã cố gắng trong việc đưa vào những kiến thức cập nhật nhưng không tránh khỏi những thiếu sót về nội dung, tính cập nhật và hình thức trình bày. Tác giả rất mong nhận được các ý kiến đóng góp từ các đồng nghiệp và độc giả để có thể hoàn thiện hơn tài liệu này.

Hà nội, 12/2014

Tác giả.

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU VỀ HỆ THỐNG THÔNG TIN QUẢN LÝ

- *Có những khác biệt giữa hệ thống thông tin quản lý và hệ thống giao dịch*
- *Các hệ thống thông tin quản lý thể hiện ở nhiều dạng khác nhau trong các giải pháp phần mềm thương mại*
- *Thiết kế hệ thống thông tin quản lý cần tập trung giải quyết vấn đề quá tải thông tin đối với người dùng*

1.1. GIỚI THIỆU CHUNG

Ngày nay, môi trường kinh doanh ngày càng cạnh tranh khốc liệt đòi hỏi các doanh nghiệp phải quản lý và khai thác các nguồn lực thông tin một cách có hiệu quả để ra các quyết định chiến lược đúng đắn và các quyết định điều hành kịp thời. Công nghệ thông tin đang làm thay đổi cách thức vận hành các tổ chức, ngành công nghiệp, các doanh nghiệp và môi trường kinh doanh. Mối quan hệ giữa khả năng ứng dụng công nghệ thông tin trong một tổ chức và khả năng thực hiện thành công các chiến lược để đạt được các mục tiêu của tổ chức đang có xu hướng trở thành mối quan hệ nhân quả và có tính phụ thuộc cao. *Hệ thống thông tin* là thể hiện cụ thể kết quả ứng dụng công nghệ thông tin trong một tổ chức. Chất lượng của hệ thống thông tin cho thấy sự hiệu quả của đầu tư ứng dụng công nghệ thông tin của một tổ chức.

Một hệ thống thông tin là một tập các thành phần có quan hệ mật thiết nhằm thu thập, xử lý, lưu trữ và phân phối thông tin nhằm hỗ trợ ra quyết định trong một tổ chức. Có thể nói hệ thống thông tin tiếp nhận *dữ liệu* như là đầu vào và xử lý chuyển đổi thành *thông tin* kết quả đầu ra có ích cho quá trình ra quyết định của tổ chức. Sự khác biệt giữa dữ liệu và thông tin cần được hiểu rõ ràng khi thiết kế các hệ hệ thống tin.

Dữ liệu được hiểu là các số liệu thô mô tả một hiện tượng cụ thể nào đó. Ví dụ, số lượng điện thoại iPhone bán ra của một cửa hàng trong một ngày nào đó, số lượng tín chỉ một sinh viên tích lũy trong một học kỳ, số lượng giảng viên có trình độ tiến sỹ, đó là dữ liệu.

Thông tin là dữ liệu có một ý nghĩa cụ thể trong một ngữ cảnh cụ thể. Ví dụ, nếu ta muốn biết một sinh viên nào đó có đủ điều kiện tốt nghiệp hay không thì số lượng tính chỉ tích lũy là thông tin còn số lượng nhân viên có trình độ tiến sĩ không phải là sinh viên. Mặt khác, nếu ta muốn biết mặt bằng học vấn chuyên môn của một trường đại học thì số lượng giảng viên có trình độ tiến sỹ là thông tin còn số lượng tín chỉ đã tích lũy của sinh viên không phải là thông tin. Như vậy, ngữ cảnh đem lại ý nghĩa cho dữ liệu và dữ liệu chuyển thành thông tin tùy thuộc vào ngữ cảnh.

Một *hệ thống thông tin* là một tập các thành phần liên hệ với nhau để thu thập, xử lý, lưu trữ và phân phối thông tin nhằm phục vụ quá trình ra quyết định của một tổ chức. Như vậy, một hệ thống thông tin sẽ thu thập dữ liệu đầu vào và xử lý tạo ra các thông tin đầu ra để phục vụ quá trình ra quyết định. Trong hệ thống thông tin hình thức, các dữ liệu và quy trình xử lý dữ liệu là có cấu trúc.

1.2. ĐỊNH NGHĨA HỆ THỐNG THÔNG TIN QUẢN LÝ

Một *hệ thống thông tin quản lý* là hệ thống thống tin cung cấp các thông tin đầu ra phục vụ quá trình ra quyết định quản lý. Ví dụ, hệ thống đặt vé máy bay, hệ thống quản lý bán hàng…

Quản lý là hoat động hoặc kỹ năng chuyển đổi các nguồn lực thành các kết quả đầu ra để hoàn thành các mục tiêu mong đợi. Hoạt động quản lý của một tổ chức liên quan đến các chức năng quản lý chính như: lập kế hoạch, tổ chức thực hiện, nhân sự, kiểm soát và thông tin. Chức năng lập kế hoạch nhằm xác định các mục tiêu và phát triển các chính sách, thủ tục và chương trình để đạt được các mục tiêu đó. Chức năng tổ chức thực hiện là việc nhóm các hoạt động và thiết lập các cơ cấu tổ chức và thủ tục đảm bảo các hoạt động được thực hiện. Chức năng nhân sự là hoạt động tuyển dụng, đào tạo các nhân viên trong tổ chức nhằm đạt được các mục tiêu và mục đích mong đợi. Chức năng kiểm soát nhằm đo lường hiệu suất so với các mục tiêu, mục đích và phát triển các thủ tục hiệu chỉnh các mục tiêu, thủ tục và hoạt động. Chức năng thông tin nhằm chuyển tải thông tin về mục tiêu, mục đích và hiệu suất tới nhân viên thông qua tổ chức và môi trường công tác.

Thông tin quản lý có thể được phân loại phục vụ quá trình ra quyết định tại các cấp quản lý khác nhau. Đối với quản lý cấp cao, các thông tin có thể được phân loại như sau:

1) Thông tin thành tích: Thông tin về tình hình hiện tại hoặc các mức thành tích đạt được so với kỳ vọng như số khách hàng được phục vụ, các mục tiêu đạt được, số bệnh nhân được điều trị, các hoạt động đã được tiến hành …
2) Thông tin trạng thái hoặc thông tin tiến độ: Thông tin hiện tại về các vấn đề, khủng hoảng và các thay đổi như tến độ xây dựng văn phòng, tình trạng nghiên cứu, thỏa thuận lao động…
3) Thông tin cảnh báo: Các thông tin về sự thay đổi mãi mãi, các sự kiện bất lợi đang xảy ra như sự sụt giảm về giá cổ phiếu, lợi nhuận, khiếu nại từ khách hàng…
4) Thông tin kế hoạch: Mô tả về các dự án, chương trình có thời hạn trong tương lại, hiểu biết về các phát triển được dự đoán như thông tin về tương lai của các nguồn tài trợ, hỗ trợ...
5) Thông tin hoạt động nội bộ: Các chỉ số về hiệu suất hoạt động của tổ chức
6) Tri thức bên ngoài: Các thông tin về ý kiến về các hoạt động trong môi trường của tổ chức. Thông tin về cạnh tranh, chính sách tài trợ, các thay đổi chính trị, các chính sách xã hội…
7) Thông tin được công bố ra bên ngoài: Báo cáo hang năm, báo cáo tiến độ quý cho nhà tài trợ, họp báo, các tài liệu công khai trước khi in ấn…

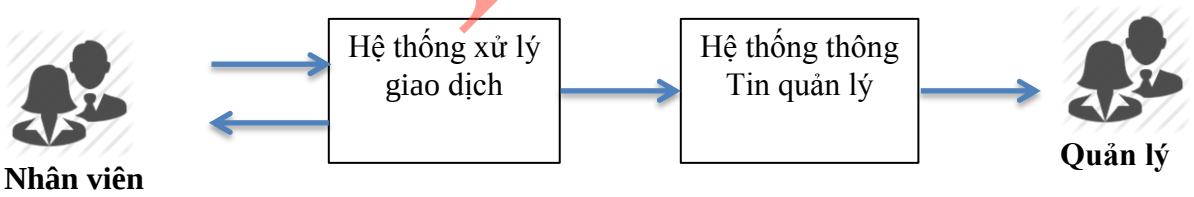
1.3. PHÂN LOẠI

Hệ thống thông thông tin của một tổ chức thường được phân làm hai loại là hệ thống hỗ trợ điều hành và hệ thống hỗ trợ quản lý. Các phân loại có mục tiêu và chức năng khác nhau.

Các hệ thống xử lý giao dịch hỗ trợ các hoạt động thường xuyên của một tổ chức hỗ trợ thu nhận và xử lý các dữ liệu từ các giao dịch nghiệp vụ. Hệ thống xử lý giao dịch thuộc loại hệ thống hỗ trợ điều hành. Dữ liệu thường được xử lý theo hai cách cơ bản: xử lý theo lô và xử lý thời gian thực. Trong hệ thống xử lý theo lô, các dữ liệu giao dịch được tích lũy theo thời gian và xử lý định kỳ. Với các hệ thống xử lý thời gian thực, dữ liệu được xử lý ngay lập tức sau khi giao dịch xảy ra. Các dữ liệu như đơn hàng, sản phẩm và kế toán được nhập vào hệ thống hàng ngày. Các dữ liệu này được sử dụng để cung cấp các thông tin cho các các nhân viên làm việc trong tổ chức. Ví dụ, phòng kinh doanh trong một công ty sẽ nhập các dữ liệu về đơn hàng vào trong hệ thống xử lý giao dịch. Các nhân viên phòng tài chính sẽ sử dụng các dữ liệu về đơn hàng để tạo ra các hóa đơn. Phòng kế toán sẽ sử dụng dữ liệu hóa đơn để cập nhật sổ cái thu chi.

Hệ thống thông tin quản lý cung cấp hỗ trợ cho việc ra các quyết định chiến thuật và chiến lược tới người quản lý và các chuyên gia nghiệp vụ. Các thông tin cung cấp bởi hệ thống này không ảnh hưởng tới các hoạt động ngắn hạn nhưng tạo cơ sở cho các quyết định dài hạn ảnh hưởng sâu rộng hơn đến các hoạt động của tổ chức. Trọng tâm của các hệ thống thông tin quản lý là tổng kết và phân tích dữ liệu giao dịch để hỗ trợ ra quyết định quản lý hiệu quả.

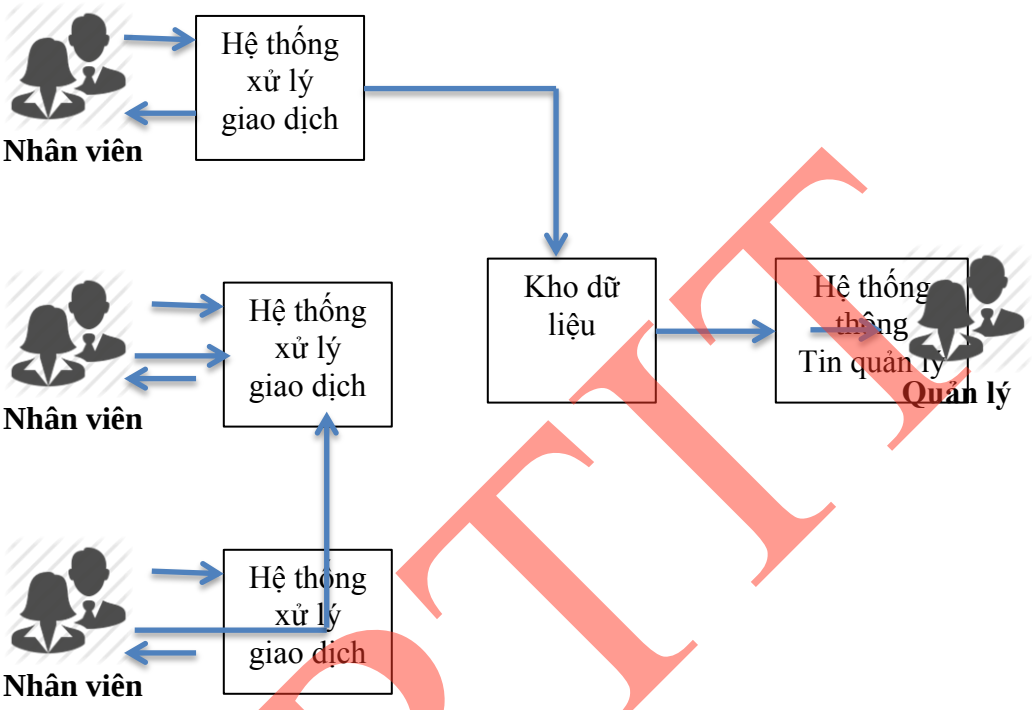
Trong các tổ chức có quy mô nhỏ, hệ thống xử lý giao dịch kiêm luôn chức năng hệ thống thông tin quản lý. Cho dù các hệ thống xử lý giao dịch không được thiết kế riêng cho mục đích quản lý, hầu hết các hệ thống xử lý giao dịch cung cấp nhiều loại báo cáo quản lý. Các nhà quản lý và nhân viên dùng chung một hệ thống. Hình 1 minh họa tổ chức hệ thống kiểu này. Cách tổ chức hệ thống kiểu này sẽ nhanh chóng bị mất kiểm soát khi kích cỡ tổ chức tăng lên vì các lý do liên quan đến đặc tính dữ liệu và đặc tính kỹ thuật. Trong trường hợp kích cỡ tổ chức tăng lên, hệ thống sẽ được sử dụng bởi nhiều



Hình 1.1: Tổ chức dùng chung hệ thống xử lý giao dịch và hệ thống thông tin quản lý.

nhân viên thường xuyên cập nhật dữ liệu dẫn đến thông tin bị thay đổi thường xuyên. Về mặt kỹ thuật, việc phân tích dữ liệu để tìm kiếm các thông tin phức tạp thường đòi hỏi năng lực tính toán của hệ thống. Dó đó, nếu người quan lý thường xuyên thực hiện các phân tích dữ liệu phức tạp sẽ gây giảm hiệu năng của hệ thống xử lý giáo dịch dùng chung.

Trong các tổ chức với dữ liệu giao dịch lớn, một tập dữ liệu mới được tạo để chuẩn bị cho hệ thống thông tin quản lý. Tập dữ liệu này có thể là dữ liệu thu nhận từ các hệ thống xử lý giao dịch trong một khoảng thời gian nhất định được kết hợp lại. Ví dụ, nếu một tổ chức có hệ thống bán hàng và hệ thống kiểm kê riêng rẽ, chúng ta có thể kết hợp dữ liệu về các đơn hàng và dữ liệu hàng hóa trong kho để xác định loại hàng nào được bán chạy và loại hàng tồn kho. Các dữ liệu trung gian dùng phục vụ hệ thống quản lý thường được tổ chức dưới dạng kho dữ liệu. Hình 2 minh họa tổ chức dữ liệu với hệ thống thông tin quản lý tách biệt với hệ thống xử lý giáo dịch



Hình 1.2: Tổ chức dùng riêng hệ thống xử lý giao dịch và hệ thống thông tin quản lý với kho dữ liệu.

Các hệ thống xử lý giao dịch và hệ thống thông tin quản lý được xây dựng theo nhiều dạng khác nhau:

- *Dạng bảng tính* ở đó dữ liệu được tổ chức trong các bảng tính với định dạng tự do hoặc dạng bảng với các hàng và cột. Dữ liệu được truy cập thông qua việc giao của hàng và cột được gọi là các ô. Dữ liệu trong các bảng tính có thể được xử lý cho các phân tích sâu hơn.
- *Cơ sở dữ liệu* cung cấp biểu diễn có cấu cấu của dữ liệu được sử dụng khi các cấu trúc dữ liệu lớn và phức tạp. Các cơ sở dữ liệu nhỏ thường được tổ chức như một phần của các bộ ứng dụng văn phòng. Khi dữ liệu giao dịch lớn, các hệ thống cơ sở dữ liệu chuyên dùng sẽ được sử dụng để lưu trữ dữ liệu với nhiều tính năng nâng cao như truy cập đa người dùng, xác thực, sao lưu dự phòng…

- *Các hệ thống báo cáo* cung cấp các thông tin quản lý thông qua các báo cáo được tạo ra từ cơ sở dữ liệu. Các hệ thống báo cáo chuyên dùng cho phép định nghĩa động các biểu mẫu báo cáo và các truy vấn cơ sở dữ liệu để đáp ứng được yêu cầu thay đổi động từ người dùng.
- *Các hệ thống tích hợp* thường được gọi là hệ thống hoạch định nguồn lực doanh nghiệp. Các gói phần mềm chức năng riêng rẽ sẽ được tích hợp nhằm chia sẻ dữ liệu và chức năng để đạt được các yêu cầu nghiệp vụ xác định.
- *Hệ thống trí tuệ doanh nghiệp*: Các hệ thống tri tuệ doanh nghiệp đại diện cho các hệ thống thông tin hàng đầu. Các hệ thống này thường làm việc trên cơ sở một tập dữ liệu từ hệ thống tích hợp. Các hệ thống này cung cấp các năng lực phân tích tích hợp và trình bày trực quan mở rộng các dữ liệu giao dịch. Một số hệ thống trí tuệ doanh nghiệp cung cấp khả năng xử lý và chuẩn bị dữ liệu.

1.4. VẤN ĐỀ THIẾT KẾ

Thiết kế nói chung là quá trình sắp xếp các phần tử của hệ thống thông tin vào một cấu trúc. Kết quả của quá trình này là một bản thiết kế hệ thống. Thiết kế hệ thống thông tin quản lý nhằm định hình các dữ liệu từ các hệ thống giao dịch vào hệ thống thông tin quản lý để hỗ trợ quá trình ra quyết định quản lý. Khái niệm thiết kế có thể được hiểu khác nhau ở các mức khác nhau bởi các chuyên gia khác nhau. Sau đây, chúng ta tìm hiểu một số cách nhìn khác nhau về thiết kế.

Các giải pháp thiết kế cho các bài toán: Sự khái niệm hóa mức cao của vấn đề thiết kế thì việc giải quyết vấn đề có thể được chia làm ba giai đoạn: tư duy, thiết kế và lựa chọn. Ở giai đoạn tư duy, người thiết kế khám và và phân tích vấn đề. Trong giai đoạn thiết kế, người thiết kế phát triển các giải pháp có thể. Trong gian đoạn lựa chọn, giải pháp phù hợp nhất sẽ được lựa chọn. Chúng ta có thể coi toàn bộ việc phát triển hệ thống thông tin quản lý là một giải pháp đối với bài toán quản lý.

Thiết kế các cấu trúc dữ liệu: Một cách nhìn hẹp hơn về vấn đề thiết kế là hoạt động phân tích các cấu trúc dữ liệu và thiết kế các mô hình dữ liệu. Thiết kế các cấu trúc dữ liệu là một phần quan trọng trong thiết kế hệ thống thông tin quản lý. Người thiết kế phải đặc biệt chú ý trong xây dựng cấu trúc các tập dữ liệu phục vụ ra quyết định quản lý.

Thiết kế các truy vấn cơ sở dữ liệu: Hoạt động thiết kế truy vấn tập trung vào việc việc tạo các truy vấn cơ sở dữ liệu nhằm lấy các dữ liệu cho người dùng. Xây dựng cấu trúc các truy vấn là một phần quan trọng trong thiết kế hệ thống thông tin quản lý.

Thiết kế các báo cáo quản lý: Hoạt động thiết kế này tập trung vào sản phẩm cuối của hệ thống thông tin quản lý là việc tạo ra các báo cáo quản lý. Hoạt động thiết kế này liên quan đến các lựa chọn thiết kế liên quan đến bố trí trực quan của dữ liệu như việc lựa chọn biểu diễn dữ liệu dưới dạng bảng hay sơ đồ.

Thiết kế chức năng hệ thống: Thiết kế chức năng hệ thống nhằm xác định các chức năng của hệ thống thông tin. Hoạt động này thước bao gồm cả hoạt động thiết kế dữ liệu được đề cấp ở phần trước.

Thiết kế cấu hình hệ thống: Xác định cấu hình phù hợp nhất của hệ thống thông tin quản lý được thực hiện ở bước thiết kế cấu hình hệ thống. Các hoạt động thiết kế cấu hình hệ thống liên quan đến các trao đổi kỹ thuật về vai trò kết xuất các dữ liệu của các hệ thống được chỉ rõ trong thiết kế kỹ thuật, thiết kế kiến trúc hoặc thiết kế hạ tầng.

1.5. VẤN ĐỀ QUÁ TẢI THÔNG TIN

Khi người dùng của hệ thống thông tin quản lý phải xử lý quá nhiều thông tin thì họ rơi vào trạng thái quá tải thông tin là trạng thái tinh thần ở đó việc cung cấp thêm thông tin cho người dùng sẽ trở thành có hại và không đem lại lợi ích cho quá trình xét đoán. Vấn đề quá tải thông tin là một chủ đề quan trọng trong thiết kế hệ thống thông tin quản lý. Các nghiên cứu trong lĩnh vực tâm lý học đã chỉ ra các giới hạn về năng lực xử lý thông tin của cong người. Ví dụ, bộ nhớ ngắn hạn của não người chỉ có thể ghi nhớ tối đa khoảng 7 kỹ hiệu khác nhau. Năng lực giới hạn trong xử lý thông tin của con người cần được xem xét cẩn thận trong thiết kế hệ thống thông tin. Một hệ thống thông tin quản lý tốt sẽ tránh cho người dùng rơi vào trạng thái quá tải thông tin khi sử dụng hệ thống được đánh giá bằng khả năng ngăn chặn cung cấp các dữ liệu không liên quan đến người dùng. Các nghiên cứu tâm lý đã chỉ ra ba vấn đề quan trọng liên quan đến giới hạn xử lý thông tin của con người mà khi thiết kế hệ thống thông tin các vấn đề này cần được xem xét cẩn trọng:

1) Khả năng quét nhanh thông tin của người dùng là giới hạn. Do đó, sự chú ý của người dùng đối với dữ liệu phụ thuộc vào yếu tố sở thích cá nhân. Như vậy, người thiết kế hệ thống không thể giả định là khi dữ liệu được trình bày đến người dùng thì chúng sẽ ngay lập tức nhận được sự chú ý từ phía người dùng.
2) Do khả năng tập trung chú ý của người dùng là hữu hạn, nên càng nhiều dữ liệu được trình bày tới người dùng thì càng ít khả năng từng phần dữ liệu sẽ nhận được sự chú ý của người dùng.
3) Càng nhiều dữ liệu được trình bày tới người dùng thì sự mất cân bằng trong sự chú ý của người dùng với các phần dữ liệu khác nhau càng tăng. Nói cách khác, phân bổ sự chú ý của người dùng với các phần dữ liệu khác nhau là không đồng đều. Ví dụ, khi nhiều dữ liệu được trình bày tới người dùng, thì người dùng thì sự chú ý của người dùng đối với phần dữ liệu đầu tiên và phần dữ liệu cuối cùng sẽ khác nhau.

Từ các đặc điểm trên, người thiết kế hệ thống thông tin cần phải thiết kế hệ thống sao cho tiết kiệm năng lực xử lý thông tin của con người và định hướng được sự chú ý của người dùng đối với thông tin trình bày thông qua cá kỹ thuật, nguyên tắc thiết kế. Các kỹ thuật tổng hợp dữ liệu và trực quan hóa thường được sử dụng để đạt được các mục tiêu này.

CHƯƠNG 2: MÔ HÌNH HÓA DỮ LIỆU VỚI MÔ HÌNH THỰC THỂ LIÊN KẾT

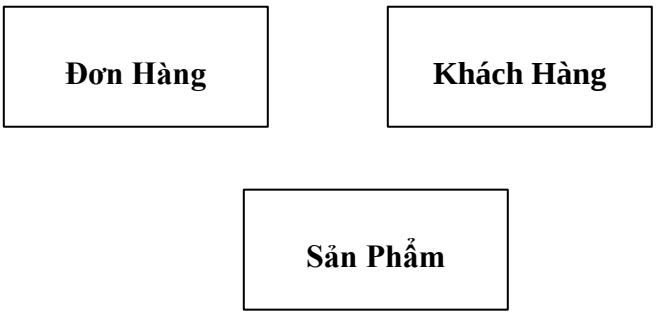
- *Mô hình hóa các thực thể của miền quản lý*
- *Mô hình hóa các quan hệ giữa các thực thể*
- *Phân loại các quan hệ giữa các thực thể*

2.1. GIỚI THIỆU MÔ HÌNH THỰC THỂ LIÊN KẾT

Trong các bài toán quản lý chúng ta cần phải có hiểu biết nhất định về các đối tượng cần được quản lý như nhân viên, khách hàng, đơn hàng v.v. Các đối tượng này là các thực thể của miền quản lý cụ thể. Quá trình thiết kế hệ thống thông tin quản lý tập trung vào phân tích tìm hiểu các thực thể cần được quản lý bởi hệ thống thông qua các phương pháp mô hình hóa dữ liệu. Các cấu trúc dữ liệu được mô hình hóa sử dụng các sơ đồ khái niệm. Kết quả của quá trình mô hình hóa dữ liệu là một tập các sơ đồ khái niệm mô tả các thực thể trong một miền quản lý được gọi là mô hình dữ liệu. Mô hình thực thể quan hệ (ER - Entity Relationship) được giới thiệu năm 1976 và vẫn được sử dụng rộng rãi ngày nay. Các thành phần chính trong mô hình thực thể quan hệ là thực thể và mối quan hệ giữa các thực thể.

2.2. THỰC THỂ, TẬP THỰC THỂ VÀ THỂ HIỆN

Thực thể được định nghĩa là một sự vật được xác định phân biệt. Ví dụ đội kinh doanh, đại lý kinh doanh, đơn hàng, khách hàng… là các thực thể. Các ví dụ cụ thể của các thực thể được gọi là các thể hiện (instance). Ví dụ, thực thể khách hàng có các khách hàng cụ thể là Nguyễn Văn A, Trần Thị B… là các thể hiện cụ thể của thực thể khách hàng. Khi mô hình hóa dữ liệu, các thực thể được xác định như là lớp chứa cho các thể hiện cụ thể. Khi mô hình hóa dữ liệu, việc xác định các thực thể không phải là công việc rõ ràng. Nhiệm vụ của người thiết kế là chuyển dịch các thể hiện cụ thể gặp phải trong miền quản lý thành các mô tả thực thể. Người thiết kế phải ra các quyết định phù hợp về việc biểu diễn các thể hiện cụ thể trong các thực thể nào. Ví dụ, anh Nguyễn Văn A vừa có thể là một thể hiện của thực thể Người, thực thể Khách Hàng và thực thể Nhân Viên. Trong sơ đồ thực thể quan hệ, các thực thể được biểu diễn bằng một hình chữ nhật với tên của thực thể ở bên trong hình chữ nhật như ở Hình 2.1



Hình 2.1: Các thực thể.

Vị trí sắp đặt của các thực thể trong sơ đồ thực thể quan hệ không mang hàm ý gì. Nói một cách khác, các thực thể có thể được đặt ở vị trí tùy ý trong sở đồ thực thể quan hệ. Thông thường, tên của thực thể thường được để ở số ít cho dù chúng ta đang mô hình hóa thực thể cho nhiều thể hiện cụ thể. Để phân biệt với cách viết tên các thuộc tính được mô tả ở phần sau, tên của thực thể thường viết bằng chữ in đậm với chữ cái đầu tiên của mỗi từ được viết hoa.

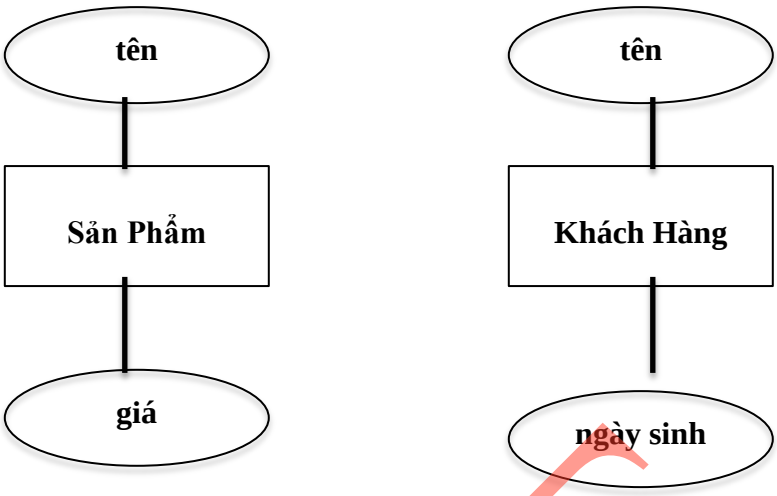
Các thể hiện của các thực thể được lưu trong các bảng đặc biệt của các cơ sở dữ liệu gọi là các *bảng quan hệ* (R-Table) hay còn gọi là *quan hệ*. Ngày nay, có rất nhiều hệ thống cơ sở dữ liệu là các cơ sở dữ liệu quan hệ được xây dựng dựa trên các cấu trúc bảng quan hệ. Trong mô hình thực thể quan hệ, mỗi thực thể tương ứng với một bảng quan hệ duy nhất. Bảng quan hệ có tên giống với thực thể trong mô hình thực thể quan hệ với một số lưu ý là: a) tên trong bảng quan hệ thường viết thường; b) không sử dụng các dấu cách giữa các từ và thường được thay thế bằng dấu gạch dưới (_). Ví dụ, tên các bảng quan hệ tương ứng với các thực thể ở hình 2.1 là: đon_hang, khach_hang, san_pham.

2.3. THUỘC TÍNH VÀ PHÂN LOẠI

2.3.1. Thuộc tính

Mỗi thực thể có các đặc tính riêng biệt được gọi là các thuộc tính của thực thể. Một *thuộc tính* là một tính chất của thực thể, một đặc tính xác định phân biệt mà tất cả các thể hiện của thực thể có thể có. Ví dụ, thuộc tính Tên của thực thể **Khách Hàng**, thuộc tính Giá của thực thể **Sản Phẩm**. Các thuộc tính của thực thể có miền giá trị xác định đặc tả thuộc tính của thực thể. Khi một thuộc tính của một thể hiện không được gán giá trị có nghĩa là thuộc tính đó bị thiếu giá trị hay có giá trị NULL.

Trong sơ đồ thực thể quan hệ, các thuộc tính được biểu diễn ký hiệu bằng các hình ellipse ở đó mỗi hình ellipse tương ứng với duy nhất một thuộc tính. Hình 2.2 minh họa ví dụ về biểu diễn các thuộc tính trong sơ đồ thực thể quan hệ.



Hình 2.2: Các thuộc tính của thực thể.

Tên của các thuộc tính được viết thường để phân biệt với tên thực thể. Việc biểu diễn các thuộc tính bằng các hình ellipse có nhược điểm là chiếm nhiều không gian. Khi số lượng thuộc tính tăng lên thì người thiết kế sẽ gặp khó khăn trong việc biểu diễn các thuộc tính. Do đó, một số các phương pháp biểu diễn thuộc tính khác được đề xuất để khắc phục nhược điểm này.

Bảng quan hệ đảm bảo biểu diễn được các thuộc tính của các thể hiện. Biểu diễn của bảng quan hệ tuân theo các luật sau:

- *Các cột*: Các cột của bảng quan hệ phải biểu diễn các thuộc tính của thực thể. Mỗi cột có tên cột để xác định cột trong bảng quan hệ. Tên của cột phải duy nhất. Thứ tự của cột trong bảng là tùy ý.
- *Các hàng*: Các cột biểu diễn một thể hiện của thực thể với mỗi hàng tương ứng với một thực thể. Các hàng không được đặt tên như các cột và thứ tự các hàng cũng tùy ý.
- *Trùng lặp các hàng*: Không được phép tồn tại hai hàng có giá trị trùng lặp hoàn toàn do mỗi hàng biểu diễn duy nhất một thực thể.
- *Các khóa chính*: Để tránh việc trùng lặp hàng xảy ra, một cột định danh được thêm vào để đảm bảo mỗi hàng được xác định duy nhất. Cột này gọi là khóa chính và thường được thiết kế có giá trị là số nguyên tự động tăng khi một hàng mới được thêm vào.

2.3.2. Kiểu thuộc tính

Các thuộc tính có các kiểu giá trị khác nhau tùy thuộc vào thông tin biểu diễn của thuộc tính. Ví dụ, tên của một sản phẩm là một sâu ký tự, năm sản xuất có thể là một số nguyên v.v. Sau đây là danh sách các kiểu dữ liệu của thuộc tính:

- *Kiểu số*: Giá trị là các số và có thể được xác định chi tiết là kiểu số nguyên, số thực. Ví dụ, số lượng sản phẩm được bán, giá của một sản phẩm.

- *Kiểu văn bản*: Giá trị là các văn bản tự do không xác định trước. Trong các hệ quản trị cơ sở dữ liệu, các kiểu *string, varchar* được sử dụng. Ví dụ, tên của khách hàng, tên của sản phẩm .
- *Kiểu phân loại*: Các giá trị không phải số, được định nghĩa trước. Ví dụ, loại sản phẩm với các giá trị Điện tử, Gia dụng, Thiết bị văn phòng…
- *Kiểu logic Đúng/Sai*: Gồm hai giá trị đúng/sai (True/False). Ví dụ, tình trạng có sẵn của sản phẩm trong kho có thể nhận hai giá trị là Đúng hoặc Sai.

Một kiểu thuộc tính đặc biệt là nhãn thời gian, nhẫn thời gian có thể là ngày hoặc thời gian ở đó một sự kiện xảy ra. Sự kiện là thuộc tính và nhãn thời gian là giá trị. Ví dụ, một đơn hàng được tạo trong một ngày cụ thể, hàng hóa được gửi vào ngày cụ thể v.v. Các nhẫn thời gian được sử dụng trong hệ thống thông tin quản lý cho các phân tích liên quan đến thời gian. Ví dụ, nhẫn thời gian có thể được sử dụng để so sách tổng khối lượng bán hàng của năm nay so với năm trước, tháng này so với tháng trước. Dó đó, thông tin thời gian cần được ghi chính xác trong hệ thống. Một số kiểu dữ liệu có thể được sử dụng cho nhãn thời gian như sau:

- *Ngày trong tuần*: Các ngày trong tuần *Thứ hai, Thứ ba, Thứ tư, Thứ năm, Thứ sáu, Thứ bảy, Chủ nhật* được biểu diễn với các kiểu giá trị phân loại hoặc bằng kiểu số
- *Ngày trong tháng*: Có giá trị nằm trong khoảng từ 1 đến 28, 29, 30 hoặc 31 tùy thuộc vào từng tháng. Các ngày trong tháng được biểu diễn bằng kiểu số.
- *Tuần trong năm*: Có giá trị nằm trong khoảng từ 1 đến 52 hoặc 53 tùy vào từng năm. Các tuần trong năm thường được biểu diễn bằng kiểu số.
- *Tháng trong năm*: Có giá trị nằm trong khoảng từ 1 đến 12. Các tháng trong năm thường được biểu diễn bằng kiểu số hoặc kiểu phân loại.
- *Năm*: Năm dương lịch
- *Thời gian*: Biểu diễn thời gian cụ thể trong ngày. Ví dụ, 7:00. Trong một vài trường hợp, giây và mili giây được sử dụng nếu cần thiết.
- *Ngày*: Kết hợp kiểu *ngày, tháng và năm*
- *Nhãn thời gian*: Kết hợp kiểu *ngày* và *thời gian*

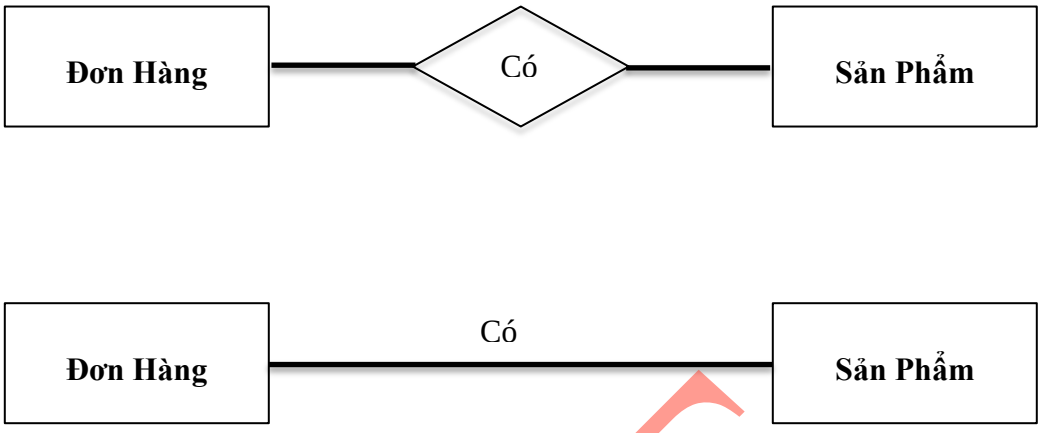
Với các nhãn thời gian, cần phải phân biệt giữa kiểu thuộc tính và định dạng thuộc tính. Định dạng thuộc tính là cách giá trị thuộc tính được trình bày tới người dùng và không quyết định cách thức giá trị thuộc tính được lưu trong hệ thống. Ví dụ về định dạng ngày ở Mỹ được trình bày là MM/DD/YYYY còn ở châu Âu, Việt Nam là DD/MM/YYYY.

2.4. MỐI QUAN HỆ VÀ PHÂN LOẠI

Các thực thể là thành phần quan trọng của mô hình dữ liệu nhưng chúng không tồn tại độc lập. Mối quan hệ giữa các thực thể đóng vai trò quan trọng và cung cấp thông tin hữu ích đến ngời dùng. Ví dụ, người quản lý muốn biết đại lý bán hàng nào được quản lý bởi tổ bán hàng nào; Nhân viên nào làm ở phòng nào. Các mối quan hệ này cần phải được mô hình hóa.

Sơ đồ ER biểu diễn một mối quan hệ bằng một đường thẳng nối hai thực thể. Các mối quan hệ có thể có tên và tên được hiển thị trong hình thoi hoặc gần đường thẳng nối hai thực thể

như ví dụ trong Hình 2.3. Các đường biểu diễn mối quan hệ có thể có các đặc trưng khác biểu diễn loại mối quan hệ. Trường hợp chỉ có đường thẳng nối hai thực thể biểu diễn loại mối quan hệ một-một ở đó mỗi thể hiện của thực thể này liên kết với một thể hiện của thực thể còn lại.



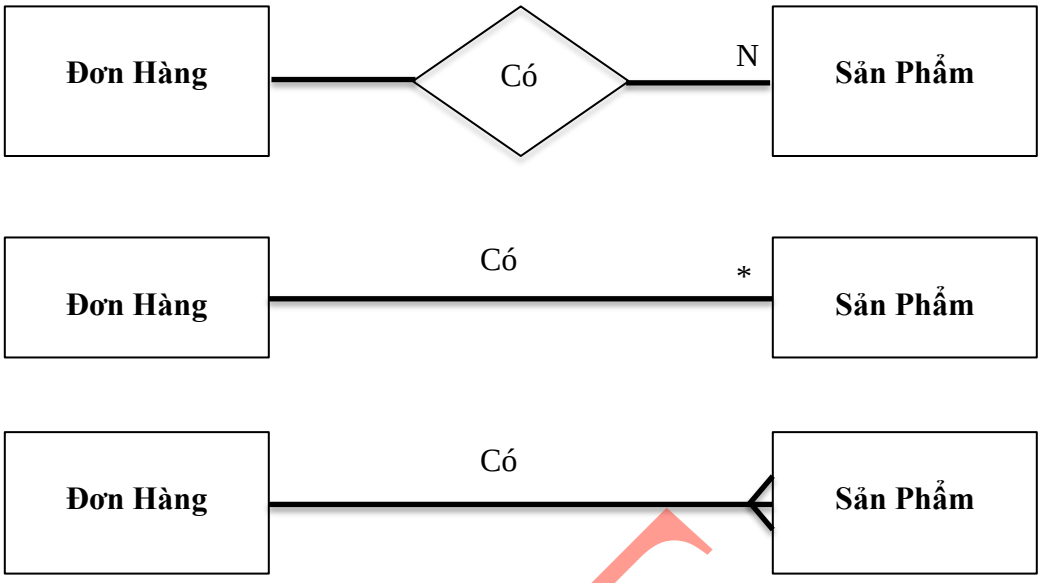
Hình 2.3. Biểu diễn quan hệ *một-một* giữa các thực thể.

Trong ví dụ về mối quan hệ giữa các thực thể Đơn Hàng và Sản Phẩm ở trên, thực tế mỗi đơn hàng có thể chứa thông tin nhiều hơn một sản phẩm. Do đó, mối quan hệ một-một là không đầy đủ trong trường hợp này. Mối quan hệ giữa các thực thể Đơn Hàng và Sản Phẩm trong ví dụ này là mối quan hệ *một-nhiều*. Như vậy, khi mỗi thể hiện của thực thể này liên kết với nhiều thể hiện của thực thể khác ta nói giữa hai thực thể có một mối quan hệ *một-nhiều*. Mối quan hệ một-nhiều được biểu diễn bằng cách thêm vào các ký hiệu ở cuối của đường thẳng nối hai thực thể ở phía đầu *nhiều*. Các ký hiệu có thể là chữ N, dấu sao (*) hoặc hình chân chim như Hình 2.4.

Để biểu diễn tùy chọn của về số lượng thể hiện của các thực thể tại các đầu của đường thẳng biểu diễn mối quan hệ giữa hai thực thể:

- Hình tròn biểu diễn *sự cho phép* số lượng thể hiện bằng không
- Nét xổ đứng biểu diễn *sự bắt buộc* số lượng thể hiện ít nhất là một.

Hình 2.5 minh họa biểu diễn các mối quan hệ với các biểu diễn về tùy chọn lực lượng của mối quan hệ.



Hình 2.4. Biểu diễn tùy chọn của quan hệ *một-nhiều* giữa các thực thể.

Biểu biểu diễn mối quan hệ trong các bảng-R, một trường thuộc tính đóng vai trò con trỏ được thêm vào bảng-R biểu diễn thực thể ở đầu *nhiều* của mối quan hệ. Trường thuộc tính con trỏ này được gọi là *khóa ngoại*.

Tính tùy chọn của mối quan hệ có liên quan đến các khóa ngoại. Nếu một mối quan hệ là tùy chọn, khóa ngoại có thể nhận giá trị NULL. Ví dụ, một bảng chứa thông tin về các đại lý bán hàng có khóa ngoại đến tổ bán hàng để biểu diễn đại lý bán hàng thuộc quản lý một tổ bán hàng. Nếu đại lý bán hàng không nhất thiết phải thuộc sự quản lý của một tổ bán hàng thì khóa ngoại có thể nhận giá trị NULL. Một ví dụ khác, một đơn hàng dứt khoát phải có ít nhất một sản phẩm, do đó bảng chứa thông tin về đơn hàng có khóa ngoại đến bảng chứa thông tin về sản phẩm và khóa ngoại này không thể nhận giá trị NULL.

Một thể hiện của A *có thể* quan hệ với không hoặc nhiều thể hiện của B

Một thể hiện của A *luôn* quan hệ với một hoặc nhiều thể hiện của B

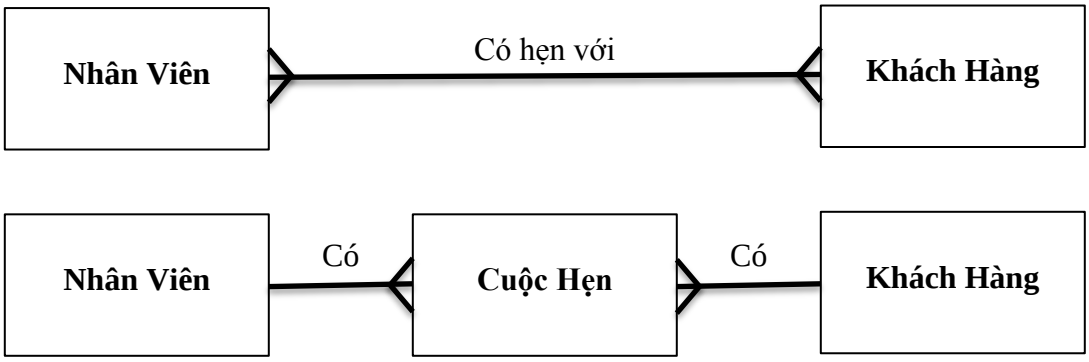
Một thể hiện của A *có thể* quan hệ với một thể hiện của B

Một thể hiện của A *luôn* quan hệ với một thể hiện của B

Hình 2.5. Biểu diễn tùy chọn lực lượng của mối quan hệ giữa các thực thể.

2.5. MỐI QUAN HỆ NHIỀU NHIỀU

Mối quan hệ *nhiều-nhiều* xảy ra khi tồn tại hai mối quan hệ *một-nhiều* ở cả hai hướng của quan hệ giữa hai thực thể. Hai thực thể có mối quan hệ nhiều-nhiều nếu mỗi thể hiện của thực thể này có liên kết với nhiều thể hiện của thực thể khác. Ví dụ mối quan hệ *giảng dạy* giữa giảng viên và môn học là mối quan hệ nhiều nhiều vì một giảng viên có thể dạy nhiều môn học và một môn học có thể được giảng dạy bởi nhiều giáo viên. Một ví dụ khác, mối quan hệ *có hẹn gặp* giữa nhân viên với khách hàng là mối quan hệ nhiều nhiều vì một nhân viên có thể có hẹn gặp với nhiều nhân viên và một khách hàng có thể có hẹn gặp với nhiều nhân viên. Mối quan hệ nhiều nhiều được biểu diễn bằng các ký hiệu nhiều (chân chim, N, hoặc dấu sao *) ở hai đầu của đường thẳng biểu diễn mỗi quan hệ. Mối quan hệ nhiều-nhiều có thể được thay thế băng hai mối quan hệ một nhiều tới một thực thể trung gian được thêm vào. Hình 2.6 là ví dụ minh họa biểu diễn mối quan hệ nhiều-nhiều giữa nhân viên và khách hàng.



Hình 2.6. Mối quan hệ nhiều-nhiều giữa các thực thể Nhân Viên và Khách Hàng.

Việc thêm một thực thể trung gian để biểu diễn mối quan hệ nhiều-nhiều có nhiều ưu điểm. Thứ nhất là nó phù hợp việc chuyển đổi mối quan hệ nhiều-nhiều từ sơ đồ ER sang các bảng-R. Thứ hai là các thuộc tính của mối quan hệ có thể được biểu diễn trong thực thể trung gian. Trong ví dụ hình 2.6, thực thể trung gian Cuộc Hẹn có thể có thêm các thuộc tính như giờ hẹn, địa điểm hẹn, trạng thái cuộc hẹn đã được xác nhận hay đã bị hủy v.v.

CHƯƠNG 3: MÔ HÌNH HÓA DỮ LIỆU VỚI UML

- *Nhận dạng sự khác biệt giữa các sơ đồ ER và các sơ đồ UML*
- *Mô hình hóa cấu trúc phân cấp thực thể thông qua tổng quát hóa*
- *Mô hình hóa vòng đời của một thực thể sử dụng sơ đồ chuyển trạng thái*
- *Chuyển đổi các cấu trúc phân cấp và các giai đoạn vòng đời thực thể thành cấu trúc bảng-R.*

3.1. GIỚI THIỆU NGÔN NGỮ MÔ HÌNH HÓA THỐNG NHẤT UML

Ngông ngữ mô hình hóa thống nhất UML sử dụng cá lớp để biểu diễn các cấu trúc dữ liệu. Lịch sử phát triển của UML gắn liền với sự phán triển của kỹ nghệ phát triển phầm mềm hướng đối tượng ở đó dữ liệu và chức năng được đóng gói trong các đối tượng. Cách tiếp cận hướng đối tượng tương phản với cách tiếp cận truyền thống ở đó dữ liệu được tách rời với chức năng.

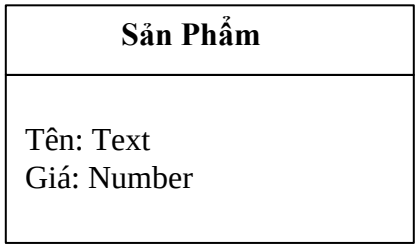
Tiếp cận hướng đối tượng là động lực cho phát triển các kỹ thuật sơ đồ hóa mới. Tuy nhiên, sự ra đời của các kỹ thuật sơ đồ hóa mới làm nảy sinh nhiều vấn đề do sự không tương thích giữa các kỹ thuật này. Ngôn ngữ mô hình hóa thống nhất UML ra đới với mục tiêu cung cấp một công cụ thống nhất cho các nhà phát triển hệ thống. Trong UML, chúng ta sẽ làm việc với các lớp và sơ đồ lớp. Khái niệm *lớp* trong mô hình UML khá tương tự với khái niệm *thực thể* trong mô hình thực thể-liên kết. UML là một trong những công cụ cơ bản để cấu trúc hóa dữ liệu.

3.2. CÁC ĐỐI TƯỢNG VÀ SỰ KẾT HỢP

Sơ đồ lớp là một phiên bản khác của sơ đồ ER và UML sử dụng thuật ngữ khác cho các thực thể và liên kết. Trong UML, thực thể được thay thế bằng lớp. Một lớp có thể chứa nhiều đối tượng nên đối tượng có thể so sánh với các thể hiện của sơ đồ ER.

Cho dù có sự khác biệt về thuật ngữ, sơ đồ lớp có rất nhiều điểm giống với sơ đồ ER. Biểu diễn đồ họa của một lớp giống với một thực thể, là hình chữ nhật. Biểu diễn một sự kết hợp cũng giống với biểu diễn liên kết trong sơ đồ ER, là một đường thẳng nối các hình chữ nhật.

Để biểu diễn các thuộc tính của các lớp, hình chữ nhật biểu diễn lợp được chia làm hai phần, phần phía trên ghi tên của lớp và phần phía dưới ghi danh sách các thuộc tính của lớp. Kiểu của thuộc tính cũng có thể được định nghĩa được tách biệt với tên thuộc tính bởi dấu hai chấm. Hình 3.1 minh họa một lớp trong UML.



Hình 3.1: Biểu diễn lớp trong UML.

Một trong những ưu điểm của biểu diễn thuộc tính của lớp trong UML so với sơ đồ ER là nhiều thuộc tính có thể được biểu diễn hiệu quả trong biểu diễn lớp.

Sơ đồ lớp sử dụng số tối đa và thối thiểu để ký hiệu lực lượng của sự kết hợp giữa hai lớp (một-một, một-nhiều). Ở cuối của biểu diễn sự kết hợp, các số tối thiểu và tối đa của lực lượng sẽ được viết. Biểu tượng dấu sao (*) chỉ thị "Nhiều" hoặc "Không giới hạn". Cách biểu diễn rút gọn lực lượng của quan hệ cũng có thể được sử dụng: Chỉ có dấu sao (*) tương đương với 0..*, chỉ có số 1 tương đương với 1..1, không có các số tương đương với 1..1 hoặc 0..1. Hình 3.2 minh họa ví dụ biểu diễn lực lượng của kết hợp trong UML.

Sơ đồ lớp UML có một loại phần tử không có trong sở đồ ER đó là *phương thức*. Một phương thức mô tả một hành vi nào đó mà một thực thể có thể thực hiện. Ví dụ, một đối tượng nhân viên kinh doanh có thể tạo đơn hàng. Hành vi tạo đơn hàng sẽ là một phương thức. Trong các hệ thống hướng đối tượng, các đối tượng tương tác với nhau bằng cách gọi các phương thức của đối tượng. Các phương thức được biểu diễn bằng thêm hình chữ nhật vào biểu diễn lớp và liệt kê tên các phương thức. Hình 3.3 minh họa một lớp với các phương thức.

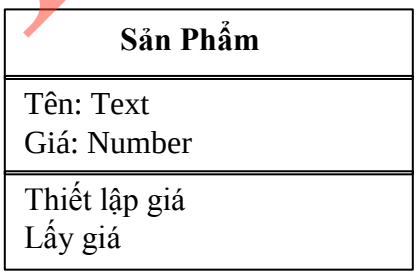
Một đối tượng của A *có thể* quan hệ với không hoặc nhiều đối tượng của B

Một đối tượng của A *luôn* quan hệ với một hoặc nhiều đối tượng của B

Một đối tượng của A *có thể* quan hệ với một thể hiện của B

Một đối tượng của A *luôn* quan hệ với một đối tượng của B

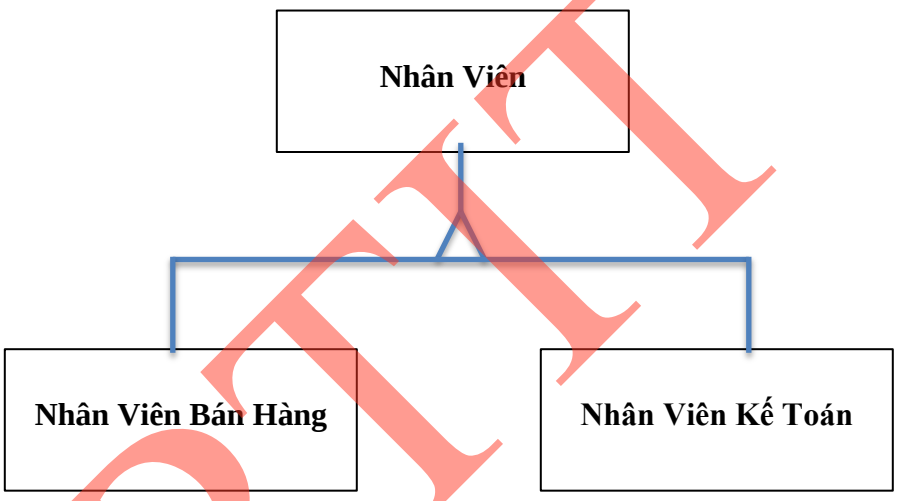
Hình 3.2. Biểu diễn tùy chọn lực lượng của mối quan hệ giữa các đối tượng.



Hình 3.3 Lớp với các phương thức.

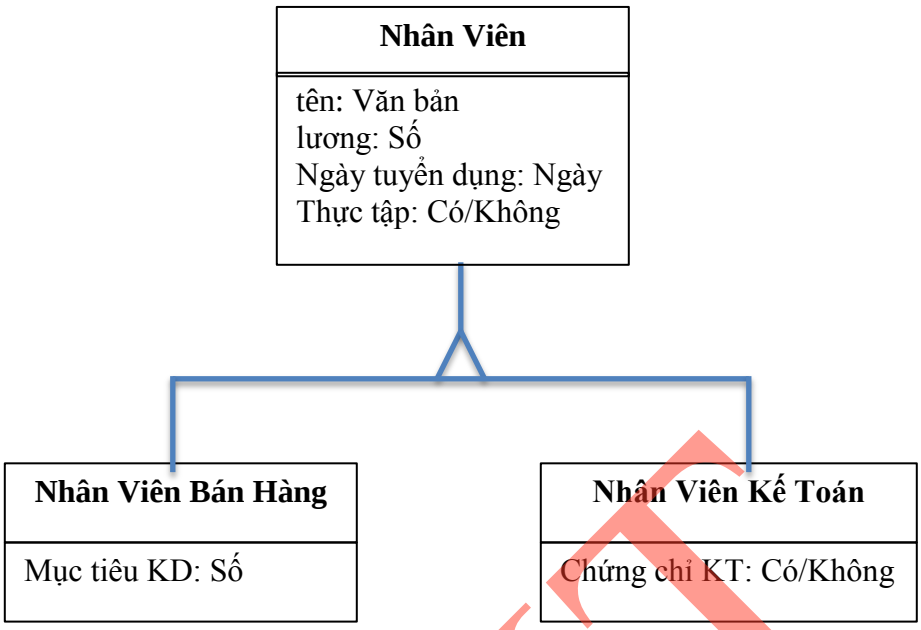
3.3. TỔNG QUÁT HÓA (GENERALIZATION)

Trong quá trình mô hình hóa, trong nhiều trường hợp các thực thể có nhiều thuộc tính chung giống nhau. Ví dụ nhân viên kinh doanh và nhân viên kế toán đều là các nhân viên nên sẽ có một số thuộc tính chung như tên, tuổi, ngày sinh v.v. Ngoài các thuộc tính chung giống nhau, các thực thể còn có các thuộc tính riêng khác nhau. Kỹ thuật tổng quát hóa của UML cho phép mô hình hóa hiệu quả các trường hợp trên. Kỹ thuật tổng quát hóa sắp xếp các lớp vào một cây ở đó các lớp tổng quát hơn sắp xếp ở trên và các lớp cụ thể được sắp xếp ở bên dưới. Hình 3.4 minh họa ví dụ tổng quát hóa.



Hình 3.4:Tổng quát hóa.

Lớp cụ thể hơn gọi là các lớp phụ (subclass) hoặc lớp con (child class) và lớp tổng quát hơn gọi là lớp cha (super class, parent class). Mỗi lớp con là một loại đặc biệt của lớp cha. Ví dụ, nhân viên bán hàng là một loại đặc biệt của nhân viên. Sử dụng kỹ thuật tổng quát hóa cho phép chúng ta xác định các thuộc tính nào thì đưa vào lớp tổng quát và thuộc tính nào được mô hình trong lớp cụ thể. Chúng ta nói rằng, các lớp con sẽ thừa kế các thuộc tính từ lớp cha do đó tránh được việc mô hình hóa các thuộc tính nhiều lần ở nhiều lớp. Hình 3.5 minh họa ví dụ về thừa kế thuộc tính.



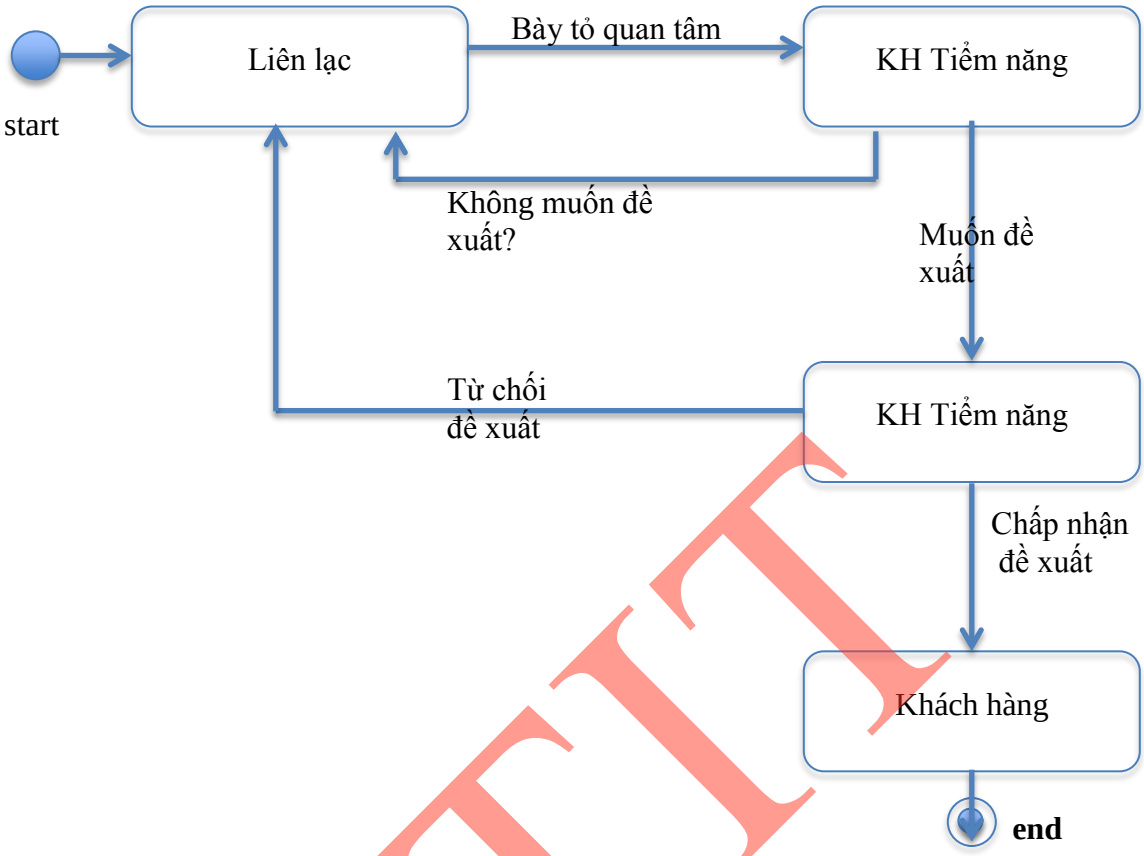
Hình 3.5 Thừa kế thuộc tính trong tổng quát hóa.

Nguyên lý đa thừa kế cho phép một lớp con kế thừa các thuộc tính từ nhiều lớp cha. Khi các lớp cha có các thuộc tính cùng tên nhưng khác kiểu dữ liệu thì cần phải có các xử lý phù hợp trong đa thừa kế.

3.4. SƠ ĐỒ CHUYỂN TRẠNG THÁI

Các thông tin về vòng đời của các thực thể đóng vai trò quan trọng trong thiết kế hướng đối tượng. Ví dụ, trong quy trình bán hàng, một khách hàng thường trải qua một số giai đoạn xác định từ xác định khách hàng tiềm năng, đến khách hàng triển vọng sau đó đến khách hàng thực sự. Người quản lý bán hảng phải quan tâm đến dịch chuyển trạng thái, ví dụ như bao nhiêu khách hàng triển vọng đã trở thành khách hàng thực sự, để có chính sách tiếp cận khách hàng phù hợp.

Một sơ đồ lớp không xem xét một cách rõ ràng các thuộc tính nào của các thực thể có thể thay đổi theo thời gian. Thực tế, các thực thể có thể thay đổi thành các thực thể khác theo thời gian. Thực thể có nhiều trạng thái và các trạng thái có các dịch chuyển, Các trạng thái và dịch chuyển có thể được biểu diễn bằng sơ đồ dịch chuyển trạng thái. Giống như sơ đồ lớp, sơ đồ dịch chuyển trạng thái là một phần của UML. Hình 3.6 mô hình dịch chuyển từ trạng thái liên lạc đến khi trở thành một khách hàng thực sự.



Hình 3.6: Biểu đồ dịch chuyển trạng thái cho lớp **Khách hàng**.

Sơ đồ dịch chuyển trạng thái bổ trợ cho sơ đồ lớp bằng cách cung cấp thông tin về hành vi động của lớp. Các sơ đồ dịch chuyển trạng thái có thể được sử dụng để diễn đạt một cách chi tiết hơn các trạng thái khác nhau mà một lớp có thể dịch chuyển qua. Có trường hợp lớp không dịch chuyển qua trạng thái nào cả, trong trường hợp đó không cần sơ đồ dịch chuyển trạng thái.

CHƯƠNG 4: TRUY VẤN DỮ LIỆU BẰNG NGÔN NGỮ SQL

- *Thảo luận cấu trúc của một truy vấn SQL*
- *Truy vấn dữ liệu từ CSDL sử dụng SQL*
- *Nối các bảng R giải chuẩn để tạo tập kết quả là cơ sở cho báo cáo quản lý*

4.1. GIỚI THIỆU NGÔN NGỮ TRUY VẤN CÓ CẤU TRÚC SQL

Trong các chương trước, chúng ta đã tìm hiểu phương pháp mô hình hóa dữ liệu quản lý. Các mô hình dữ liệu sẽ được cài đặt lưu trữ trong các bảng quan hệ là nền tảng bên dưới của hệ thống thông tin quản lý.

Quá trình lấy dữ liệu từ các bảng gọi là truy vấn. Chương này sẽ giải thích các vấn đề cơ bản của truy vấn dữ liệu. Truy vấn dữ liệu được thực hiện thông qua cung cấp các chỉ dẫn tới hệ quản trị cơ sở dữ liệu. Các câu lệnh được xây dựng dử dụng một ngôn ngữ thao tác với các bảng dữ liệu là ngông ngữ hỏi có cấu trúc SQL. Ngoài việc nhập trực tiếp các câu lệnh SQL, chúng ta có thể sư dụng các công cụ trực quan để xây dựng truy vấn, các công cụ này có nhiệm vụ chuyển đổi chỉ dẫn của người dụng dưới dạng trực quan thành các phát biểu SQL.

4.2. PHÉP CHỌN CÁC THUỘC TÍNH

SQL bao gồm một tập các câu lệnh được định nghĩa trước. Một câu lệnh SQL được tạo thành từ các phần khác nhau gọi là các mệnh đề. Câu lệnh SELECT cho phép lấy các dữ liệu từ một hoặc nhiều bảng. Ví dụ một bảng quan hệ lưu dữ liệu của sản phẩm trong bảng **san_pham** như sau:

ma_san_pham	ten	gia
1	Cam	1000
2	Xà phòng	50000
3	Rượu	70000
4	Táo	2000
5	Nến	8000

Ví dụ một bảng quan hệ lưu dữ liệu về nhân viên bán hàng trong bảng **nhan_vien_bh** như sau:

ma_nv	ten	thuc_tap
=====	========	=======
1	Nam	true
2	Tuấn	true
3	Hương	false
4	Thủy	false
5	Minh	false

Giả sử chúng ta muốn lấy dữ liệu từ một thuộc tính của bảng SAN_PHAM. Chúng ta muốn hiển thị tất cả các tên sản phẩm đang bán sử dụng câu lệnh SELECT như sau:

```
SELECT    ten
FROM      san_pham
```

Như vậy, chúng ta chỉ cần chỉ rõ tên của thuộc tính chúng ta muôn lấy dữ liệu trong mệnh đề của câu lệnh SELECT và tên của bảng từ đó chúng ta lấy dữ liệu trong mệnh đề FROM. Các từ khóa SELECT và FROM là bắt buộc cho mọi câu lệnh SELECT.

Với câu lệnh trên, hệ quản trị CSDL sẽ trả về một danh sách các hàng gọi là tập kết quả như sau:

```
ten
=======
Cam
Xà phòng
Rượu
Táo
Nến
```

Tên cột trong tập kết quả mặc định là tên của thuộc tính. Chúng ta có thể đổi tên cột sử dụng mệnh đề AS như sau:

```
SELECT    ten    AS    Product
FROM      san_pham
```

Ta sẽ được kết quả như sau:

```
Product
=======
Cam
Xà phòng
Rượu
Táo
Nến
```

Bộ phân tích cú phát SQL sẽ không thể diễn dịch một câu lệnh SQL nếu tên của bảng quan hệ hoặc tên thuộc tính có chứa khoảng trống. Do đó, sử dụng dấu gạch dưới thay cho khoảng trống khi đặt tên bảng và thuộc tính được sử dụng phổ biến để tránh các lỗi liên quan đến khoảng trắng. Các câu lệnh SQL có thể được viết trên một dòng. Tuy nhiên, để trình bày rõ ràng, thông thường mốt mệnh đề của câu lệnh thường được trình bày trên mộ dòng.

Trong ví dụ tiếp theo, chúng ta mong muốn chọn nhiều hơn một thuộc tính từ một bảng. Chúng ta có thể liệt kê các thuộc tính muốn lấy ra sau mệnh đề SELECT. Giả sử, chúng ta muốn lấy tên và giá của tất cả các sản phẩm từ bảng **san_pham**, câu lệnh SQL như sau:

```
SELECT    ten, gia
FROM      san_pham
```

Kết quả của câu lệnh trên là:

```
ten          gia
=======      ======
Cam          1000
Xà phòng     50000
Rượu         70000
Táo          2000
Nến          8000
```

Trong trường hợp chọn nhiều hơn một thuộc tính, trình tự của các thuộc tính trong mệnh đề SELECT sẽ quyết định trình tự dữ liệu của các thuộc tính tương ứng được trả về trong tập kết quả.

Nếu chúng ta muốn lấy tất cả các thuộc tính trong một bảng, chúng ta dùng dấu sao sau mệnh đề SELECT như ví dụ như sau:

```
SELECT *
FROM nhan_vien_bh
```

Câu lệnh trên sẽ lấy tất cả các thông tin của nhân viên bán hàng từ bảng **nhan_vien_bh**.

Thứ tự các hàng trả về trong tập kết quả mặc định là thứ tự theo thuộc tính khóa. Trong trường hợp chúng ta muốn sắp xếp thứ tự các hàng dựa trên thuộc tính khác, chúng ta có thể sử dụng mệnh đề ORDER BY trong câu lệnh SELECT. Ví dụ, nếu chúng ta muôn có một danh sách các sản phẩm và giá được sắp xếp theo thứ tự giá, chúng ta sử dụng câu lệnh SQL sau:

```
SELECT      ten, gia
FROM        san_pham
ORDER BY    gia
```

Kết quả như sau:

```
 ten          gia
=======      ======
Cam          1000
Táo          2000
Nến          8000
Xà phòng     50000
Rượu         70000
```

Chúng ta có thể sắp xếp các hàng trả về dựa trên nhiều hơn một thuộc tính trong mệnh đề ORDER BY. Việc sắp xếp sẽ thực hiện theo thứ tự ưu tiên liệt kê trong mệnh đề. Mặc định, thứ tự sắp xếp từ thấp đến cao, trong trường hợp muốn sắp xếp từ cao xuống thấp, chúng ta sử dụng thêm từ khóa DESC. Ví dụ, nếu chúng ta lấy tất cả các sảng phẩm và sắp xếp kết quả theo giá giảm dần:

```
SELECT      ten, gia
FROM        san_pham
ORDER BY    gia      DESC
```

Kết quả như sau:

```
 ten          gia
=======      ======
Rượu         70000
Xà phòng     50000
Nến          8000
Táo          2000
Cam          1000
```

4.3. PHÉP CHỌN CÓ ĐIỀU KIỆN

Câu lệnh SELECT sẽ lấy tất cả các hàng trong bảng quan hệ. Trong đa số các trường hợp, chúng ta quan tâm đến một tập con các hàng. Để chọn tập con các hàng, chúng ta có thể sử dụng mệnh đề điều kiện. Nếu một hàng thỏa mãn điều kiện thì thực thể đó sẽ được trả về trong tập kết quả và ngược lại. Các điệu kiện được biết đến dưới dạng các biểu thức Boolean được thêm vào sau mệnh đề WHERE của câu lệnh SQL. Mệnh đề WHERE này đứng đằng sau mệnh đề FROM và trước mệnh đề ORDER BY. Ví dụ:

```
SELECT      ten, gia
FROM        san_pham
WHERE       gia >5000
ORDER BY    gia
```

Câu lệnh trên sẽ có kết quả sau:

```
ten         gia
=======     ======
Nến         8000
Xà phòng    50000
Rượu        70000
```

Khi xây dựng các biểu thức, chúng ta sẽ gặp phải một số trường hợp đặc biệt. Với các biểu thức áp dụng cho các thuộc tính kiểu số thì sẽ không gặp vấn đề gì. Chúng ta chỉ cần chỉ rõ điều kiện thuộc tính bằng, nhỏ hơn, hoặc lớn hơn .v.v một giá trị cụ thể. Với thuộc tính có kiểu xâu ký tự (text), các điều kiện thường bị giới hạn bởi so sánh giống và khác nhau.

Các thuộc tính kiểu ngày tháng (Date) hoặc thời gian (Time) cũng thường được sử dụng trong biểu thức Boolean. Các toán tử 'bằng', 'lớn hơn' và 'nhỏ hơn' có ý nghĩa về thời gian tương ứng là 'tại thời điểm khi', 'sau ' và 'trước'. Do đó, biểu thức date >'1 Jan 2000' có nghĩa là 'sau ngày mồng 1 tháng 1 năm 2000'.

Kiểu thuộc tính Đúng/Sai đóng vai trò như một biểu thức Boolean. Ví dụ, câu truy vấn sau:

```
SELECT ten, thuc_tap
FROM  nhan_vien_bh
WHERE       thuc_tap
```

sẽ lựa chọn tất cả các nhân viên bán hàng đang trong giai đoạn thực tập.

Chúng ta có thể ghép các biểu thức Boolean với nhau sử dụng từ khóa AND và OR. Từ khóa AND hàm ý tất cả các điều kiện cần phải đúng thì hàng dữ liệu mới được trả về trong tập kết quả. Từ khóa OR hàm ý rằng bất kỳ điều kiện nào đúng thì hàng dữ liệu sẽ được trả về trong tập kết

quả. Các biểu thức nên được đặt trong ngoặc đơn khi sử dụng với các từ khóa AND, OR. Giả sử dúng ta có bảng chi tiết đơn hàng **chi_tiet_don_hang** như sau:

tt_dh	ma_dh	ma_sp	so_luong
1	1	1	50
2	1	3	50
3	2	1	10
4	3	4	10
5	3	1	75

Sử dụng từ khóa AND, chọn tất cả các chi tiết đơn hàng liên quan đến táo (ma_sp = 1) với số lượng lớn hơn 25 như sau:

```
SELECT      *
FROM        chi_tiet_don_hang
WHERE       (ma_sp = 1) AND (so_luong > 25)
```

Kết quả sẽ là:

tt_dh	ma_dh	ma_sp	so_luong
1	1	1	50
5	3	1	75

Dùng từ khóa OR, chúng ta có thể lấy ra các đơn hàng liên quan đến táo hoặc đơn hàng có số lượng lớn hơn 25 như sau:

```
SELECT      *
FROM        chi_tiet_don_hang
WHERE       (ma_sp = 1) OR (so_luong > 25)
```

Kết quả như sau:

tt_dh	ma_dh	ma_sp	so_luong
1	1	1	50
2	1	3	50
3	2	1	10
5	3	1	75

Khi chúng ta muốn diễn đạt các điều kiện ở dạng phủ định, chỉ cần đặt từ khóa NOT trước biểu thức. Ví dụ:

```
SELECT      *
FROM chi_tiet_don_hang
WHERE NOT (ma_sp = 1)
```

Câu lệnh trên sẽ chọn tất cả các đơn hàng không phải sản phẩm là táo. Tập kết quả trả về như sau:

tt_dh	ma_dh	ma_sp	so_luong
====	===== =====	=======	
2	1	3	50
4	3	4	10

4.4. PHÉP KẾT NỐI CÁC BẢNG QUAN HỆ

Trong hệ thống thông tin quản lý, khi tạo các báo cáo tổng hợp chúng ta thường phải tổng dữ liệu liên quan từ nhiều bảng. Phép nối cho phép nối các bảng dựa trên các thuộc tính chung.

Xem xét các bảng dữ liệu nhóm bán hàng **nhom_bh** và nhân viên bán hàng **nhan_vien_bh**. Đây là quan hệ một-nhiều do mỗi nhóm bán hàng có thể có nhiều nhân viên bán hàng. Bảng **nhan_vien_bh sẽ** có một khóa ngoại là mã của nhóm bán hàng. Mã của nhóm bán hàng là khóa chính trong bảng nhóm bán hàng. Quan hệ giữa hai bảng này là quan hệGiả sử chúng ta muốn hiện thị danh sách các nhân viên bán hàng và tên của các nhóm của họ. Trước tiên, chúng ta liệt kê các bảng riêng rẽ.

```
SELECT      ma_nv, ten, ma_nhom
FROM        nhan_vien_bh
ORDER BY    ma_nv
```

Kết quả có được như sau:

ma_nv	ten	ma_nhom
=====	========	=======
1	Nam	1
2	Tuấn	2
3	Hương 1	
4	Thủy	2
5	Minh	NULL

và

```
SELECT      ma_nhom, ten
FROM        nhom_bh
```

```
ORDER BY    ma_nhom
```

Kết quả là:

```
ma_nhom     ten
=======     =======
1           Alpha
2           Beta
3           Gamma
```

Bây giờ chúng ta muôn kết hợp hai bảng này bắt đầu với việc kết hợp tên của nhóm bán hàng với tên của nhân viên bán hàng. Giả sử dùng câu lệnh như sau:

```
SELECT      ten, ten
FROM        nhan_vien_bh, nhom_bh
```

Câu lệnh trên sẽ gây nhầm lẫn cho trình phân tích cú pháp SQL do không thể xác định thuộc tính **ten** nào thuộc về bảng nào. Trình phân tích cú pháp SQL sẽ báo lỗi liên quan đến sự tối nghĩa của tên thuộc tính. Giải pháp có thể là lấy tên bảng làm tiền tố cho các thuộc tính như sau:

```
SELECT      nhan_vien_bh.ten, nhom_bh.ten
FROM        nhan_vien_bh, nhom_bh
```

Chúng ta sẽ có kết quả trả về như sau

```
nhan_vien_bh.ten        nhom_bh.ten
=====                   =======
Nam                     Alpha
Nam                     Beta
Nam                     Gamma
Tuấn                    Alpha
Tuấn                    Beta
Tuấn                    Gamma
Hương                   Alpha
Hương                   Beta
Hương                   Gamma
Thủy                    Alpha
Thủy                    Beta
Thủy                    Gamma
Minh                    Alpha
Minh                    Beta
Minh                    Gamma
```

Trình phân tích cú pháp SQL có kết hợp tất cả các khả năng các hàng thuộc hai bảng.

Chúng ta thay đổi câu lệnh truy vấn trên bằng cách lựa chọn thêm một số thuộc tính khác từ 2 bảng như sau:

```
SELECT      nhan_vien_bh.ten, nhan_vien_bh.ma_nhom, nhom_bh.ma_nhom nhom_bh.ten
FROM        nhan_vien_bh, nhom_bh
```

Kết quả trả về như sau:

nhan_vien_bh.ten	nhan_vien_bh.ma_nhom	nhom_bh_ma_nhom	nhom_bh.ten
=====	=======	================	==========
Nam	1	1	Alpha
Nam	1	2	Beta
Nam	1	3	Gamma
Tuấn	2	1	Alpha
Tuấn	2	2	Beta
Tuấn	2	3	Gamma
Hương	1	1	Alpha
Hương	1	2	Beta
Hương	1	3	Gamma
Thủy	2	1	Alpha
Thủy	2	2	Beta
Thủy	2	3	Gamma
Minh	NULL	1	Alpha
Minh	NULL	2	Beta
Minh	NULL	3	Gamma

Quan sát dữ liệu ở kết quả trên chúng ta thấy có một số hàng mã nhóm của hai bảng là giốn nhau. Chúng ta sử dụng thông tin này để tạo câu lệnh sau:

```
SELECT      nhan_vien_bh.ten, nhom_bh.ten
FROM        nhan_vien_bh, nhom_bh
WHERE       nhan_vien_bh.ma_nhom = nhom_bh.ma_nhom
```

Kết quả có được từ câu lênh trên như sau:

```
nhan_vien_bh.ten        nhom_bh.ten
=====                   =======
Nam                     Alpha
Tuấn                    Beta
Hương           Alpha
Thủy                    Beta
```

Chúng ta có thể thay đổi trình tự thuộc tính và thực thể để có báo cáo dễ nhìn hơn bằng cách nhóm theo nhóm bán hàng như sau:

```
SELECT      nhom_bh.ten, nhan_vien_bh.ten
FROM        nhan_vien_bh, nhom_bh
WHERE       nhan_vien_bh.ma_nhom = nhom_bh.ma_nhom
ORDER BY    nhom_bh.ten, nhan_vien_bh.ten
```

Kết quả có được từ câu lệnh trên như sau:

```
nhom_bh.ten             nhan_vien_bh.ten
=====                   =======
Alpha                   Nam
Alpha                   Hương
Beta                    Tuấn
Beta                    Thủy
```

Trong ví dụ trình bày ở phần này, chúng ta đã dùng các khóa chính và khóa ngoại trong một quan hệ một-nhiệu để nối các thuộc tính trong hai bảng. Đâu tiên, lựa chọn các thuộc tính từ hai bảng. Sau đó dùng mệnh đề WHERE để kiểm tra điều kiện bằng nhawu của khóa ngoại thuộc một bảng với khóa chính thuộc bảng còn lại. Sau đó sắp xếp lại thứ tự các thuộc tính cho phù hợp với yêu cầu trình bày báo cáo.

Ngôn ngữ SQL cung cấp mệnh đề đặc biệt JOIN được sử dụng để nối hai bảng đảm bảo rằng dữ liệu từ mối quan hệ tùy chọn cũng được lựa chọn. Có nhiều kiểu nối như nối trong INNER JOIN, nối trái LEFT JOIN và nối phải RIGHT JOIN. Chi tiết các mệnh đề nối này đã được trình bày trong tài liệu môn học CSDL căn bản

4.5. ĐA KẾT NỐI VÀ GIẢI CHUẨN DỮ LIỆU

Trong phần trước, chugns ta đã tìm hiểu việc nối hai bảng với quan hệ một-nhiều. Trong nhiều trường hợp, một bảng có chứa nhiều hơn một khóa ngoại và có thể thực hiện phép nối trên nhiều hơn hai bảng. Cụ thể, trong các quan hệ nhiều-nhiều, một bảng trung gian được sử dụng để biểu diễn mối quan hệ nhiều-nhiều và chứa nhiều hơn một khóa ngoại. Ví dụ, bảng dữ liệu về chi tiết đơn hàng **chi_tiet_don_hang** tạo quan hệ giữa đơn hàng và sản phẩm và đây là quan hệ nhiều-nhiều. Do đó, nếu muốn hiển thị thông tin về chi tiết đơn hàng sẽ cần dữ liệu từ bảng lưu dữ liệu đơn hàng và bảng lưu dữ liệu về sản phẩm. Trong ví dụ này, nếu chúng ta nối bảng dữ liệu đơn hàng và bảng dữ liệu nhân viên bán hàng để hiện thị tên nhân viên bán hàng trong các đơn hàng, thì cũng có thể hiển thị khóa ngoại của nhóm bán hàng mà nhân viên bán hàng thuộc nhóm đó. Do đó, nếu chúng ta muốn hiển thị thêm thông tin về tên của nhóm bán hàng thì cần phải thực hiện phép nối khác với bảng dữ liệu chứa thông tin về nhóm bán hàng.

Khi thực hiện nối các thuộc tính thuộc nhiều bảng, tập kết quả có các thuộc tính phi định danh liên quan đến bảng khởi tạo. Quá trình này gọi là quá trình giải chuẩn dữ liệu. Trong quá trình giải chuẩn, tất cả các khóa ngoại bị loại bỏ từng bước một. Danh sách sau đây trình bày cách thức quá trình giải chuẩn hoạt động và áp dụng vào ví dụ bảng dữ liệu chi tiết đơn hàng được giải chuẩn.

Bắt đầu với bảng khởi tạo: Bắt đầu với bảng chi_tiet_don_hang. Bảng này chứa các khóa ngoại đến bảng dữ liệu về đơn hàng và sản phẩm.

Các phép nối bậc nhất: Trước tiên, loại bỏ khóa ngoại ma_dh bằng phép nối với các thuộc tính của bảng dữ liệu đơn hàng. Bảng kết quả sẽ dư thừa dữ liệu và chứa hai khóa ngoại mới cho khách hàng và nhân viên.

Tiếp theo, loại bỏ khóa ngoại thứ hai là ma_sp bằng phép nối với bảng san_pham để nhận được các thuộc tính cho các sản phẩm như là tên sản phẩm, giá sản phẩm. Bảng kết quả nhận thêm một khóa ngoại tham chiếu đến loại sản phẩm.

Các phép nối bậc hai: Các phép nối này loại bỏ các khóa ngoại xuất hiện khi thực hiện các phép nối bậc nhất các bảng lân cận. Trong ví dụ xem xét, liên quan đến việc nối với các bảng khach_hang, nhan_vien_bh, loai_san_pham. Khi nối các bảng này sẽ phát sinh thêm các khóa ngoại mới là ma_nhom và ma_trang_thai.

Các phép nối bậc 3: Thực hiện nối nhóm bán hàng và trạng thái khách hàng với bảng chi tiết đơn hàng sẽ được kết quả cuối cùng không có khóa ngoại trong tập kết quả.

Giải chuẩn cho phép loại bỏ các khóa ngoại và cho tập kết quả với các thuộc tính phi định danh kết hợp với bảng ban đầu. Bảng dữ liệu kết quả sau quá trình giải chuẩn là cơ sở xây dựng báo cáo trong hệ thống thông tin quan lý. Kho dữ liệu là một tập lựa chọn các bảng dữ liệu được giải chuẩn từ các bảng dữ liệu chuẩn hóa của các hệ thống xử lý giao dịch. Các bảng dữ liệu trong kho dữ liệu có thể được xem là kết quả của quá trình tiền xử lý dữ liệu sẽ được sử dụng bởi hệ thống thông tin quản lý. Tiền xử lý là cần thiết vì các vấn đề liên quan đến hiệu năng của hệ

thống. Các bảng dữ liệu chuẩn hóa cũng được dùng trong các hệ thống thông tin quản lý và được tham chiếu đến như là dữ liệu thô cho báo cáo quản lý.

CHƯƠNG 5: TỔNG HỢP DỮ LIỆU

- *Hiểu các xử lý cơ bản trên tập kết quả đã giải chuẩn*
- *Các phương pháp tổng hợp dữ liệu*
- *Liên hệ các lựa chọn khác nhau để tổng hợp dữ liệu với các loại thang đo khác nhau*
- *Xây dựng các bảng tổng hợp, các bảng cross-tab, pivot sử dụng các xử lý trên tập kết quả và các lựa chọn tổng hợp*

5.1. TẦM QUAN TRỌNG CỦA VIỆC TỔNG HỢP DỮ LIỆU

Dữ liệu tổng hợp liên quan đến vieecjt tạo ra dữ liệu về các nhóm của các hàng dữ liệu. Thông tin về tổng lợi tức trên các đơn hàng, sản lượng bán trung bình, số lượng bán tối đa một sản phẩm trong một tháng nào đó… là các ví dụ về dữ liệu tổng hợp. Có thể nói rằng, tổng hợp dữ liệu một cách hiệu quả là một kỹ năng thiết yếu của một nhà quản lý. Do đó, một hệ thống thông tin quản lý phải có khả năng hỗ trợ tổng hợp dữ liệu một cách nhanh chóng, chính xác và linh hoạt.

Có nhiều để tạo ra dữ liệu tổng hợp. Sử dụng ngôn ngữ truy vấn SQL để lựa chọn thông tin về các nhóm thực thể. Ví dụ, sử dụng mệnh đề GROUP BY cũng như các hàm tổng hợp như SUM, AVG, MAX v.v. Sử dụng các ứng dụng bảng tính hoặc các gói phần mềm cho trí tuệ doanh nghiệp, kho dữ liệu là cách tiếp cận ngày càng phổ biến trong việc tạo ra dữ liệu tổng hợp. Các công cụ bảng tính hoặc phần mềm trí tuệ doanh nghiệp chuyên dùng thường cung cấp các chức năng tổng hợp dữ liệu nâng cao. Các công cụ này thường cũng được trang bị các kỹ thuật trình bày trực quan mạnh mẽ. Mặc dù các cơ sở dữ liệu là công cụ quan trọng trong thu thập tổ chức dữ liệu giao dịch. Xử lý các dữ liệu tổng hợp sẽ liên quan nhiều đến các công cụ bảng tính và trí tuệ doanh nghiệp.

5.2. THAO TÁC VỚI CÁC BẢNG TỔNG HỢP

Trong chương trước, chúng ta đã tìm hiểu quy trình giải chuẩn để tạo ra các bảng dữ liệu được giải chuẩn chữa tập các dữ liệu giao dịch. Các bảng đó biểu diễn các tập hợp các thuộc tính từ các bảng quan hệ được nối với nhau trong các bản lớn để phục vụ phân tích dữ liệu. Các bảng dữ liệu được giải chuẩn trung gian đóng vai trò sống còn trong việc sinh ra các thông tin quản lý đầy đủ với các thuật ngữ cụ thể liên quan đến các loại bảng này.

Các cột trong các bảng được giải chuẩn được gọi là *các biến*. Mỗi biến có một miền giá trị và đại diện cho một *chiều của dữ liệu*. Đây chính là lý do tại sao các bảng này cũng được gọi là các bảng đa chiều ở đó các chiều là các cột của bảng. Việc các cột này được biểu diễn ban đầu như là các thuộc tính trong các bảng quan hệ không cần phải được quan tâm ở đây. Các bảng được giải chuẩn thường rất lớn và chứa một số lớn các cột và hàng. Để hợp lý hóa dữ liệu, các

bảng này phải được rút gọn thành một dạng ý nghĩa hơn để dễ phân tích và quá trình rút gọn này gọi là xử lý bảng. Một số thuật ngữ liên quan đến cách thức xử lý các bảng.

Thuật ngữ được sử dụng cho việc chọn một tập con các hàng trong một bảng đa chiều gọi là *cắt (slicing)*. Chúng ta 'cắt' bảng bằng cách loại bỏ các hàng mà chúng ta không quan tâm. Trong ví dụ về quản lý bán hàng, chúng ta chẳng hạn chỉ chọn các dữ liệu giao dịch liên quan đến nhóm Alpha.

Thuật ngữ sử dụng cho việc chọn một số các cột (chiều) của bảng dữ liệu đa chiều gọi là *xắt (dicing)*. Chúng ta 'xắt' một bảng bằng cách loại bỏ các cột không quan tâm. Ví dụ, chúng ta chỉ quan tâm đến tên của các nhân viên bán hàng trong một số phân tích hiệu quả bán hàng của nhân viên.

Với hai hoạt động trên cho phép chúng ta 'cắt và xắt' các bảng đa chiều thành các bảng nhỏ hơn phù hợp hơn với các mục đích cụ thể. Hình 5.1minh họa hai hoạt động 'cắt' và 'xắt'. Trong ví dụ ở hình 5.1, hoạt động cắt loại bỏ các hàng không chứa giá trị A ở cột đầu tiên. Hoạt động xắt loại bỏ các cột ta không quan tâm.

Loại xử lý thứ ba và thứ tư trên các bảng liên quan đến tổng hợp dữ liệu. Chúng ta có thể co một số hàng bằng cách thay thê bằng một hàng với các giá trị tổng hợp đại diện. Ví dụ, chúng ta có bảng dữ liệu như bảng 5.1 và co bảng đó lại bằng việc nhóm the mã đơn hàng như bảng 5.2 hoặc nhóm theo sản phẩm như bảng 5.3

	Cột 1	Cột 2	Cột 3	Cột 4
Hàng 1	A	C	…	…
Hàng 2	A	D	…	…
Hàng 3	B	C	…	…
Hàng 4	B	D	…	…

Xắt (dicing)

	Cột 1	Cột 2
Hàng 1	A	C
Hàng 2	A	D
Hàng 3	B	C
Hàng 4	B	D

Cắt (slicing)

Cắt và xắt (Slicing and dicing)

	Cột 1	Cột 2	Cột 3	Cột 4
Hàng 1	A	C	…	…
Hàng 2	A	D	…	…

	Cột 1	Cột 2
Hàng 1	A	C
Hàng 2	A	D

Hình 5.1 Các hoạt động cắt và xắt (Slicing and dicing).

Bảng 5.1 Bảng dữ liệu giải chuẩn mẫu với dữ liệu giao dịch

Don_hang	Khach_hang	San_pham	Gia	So_luong
1	Hương	Cam	1	6
1	Hương	Nến	7	34
2	Loan	Rượu	10	24
2	Loan	Cam	1	4
2	Loan	Táo	1	35

Bảng 5.2 Bảng dữ liệu được nhóm theo mã đơn hàng

Don_hang	Khach_hang	San_pham	Gia	So_luong
1	Hương	Cam	4	40
2	Loan	Táo	4	63

Bảng 5.3 Bảng dữ liệu giải được nhóm theo sản phẩm

Don_hang	Khach_hang	San_pham	Gia	So_luong
-	-	Cam	1	10
-	-	Nến	7	34
-	-	Rượu	10	24
-	-	Táo	1	35

Quá trình gộp dữ liệu thành các giá trị tổng hợp được gọi là cuộn dữ liệu lên (rolling up). Quá trình ngược lại mở rộng giá trị dữ liệu tổng hợp thành các giá trị giao dịch gọi là quá trình đào sâu dữ liệu (drilling down).

Như vậy, có bốn hoạt động xử lý trên bảng dữ liệu được giải chuẩn là cắt, xắt, cuộn lên và đào sâu. Các gói phần mềm trí tuệ doanh nghiệp thường hỗ trợ thực hiện các hoạt động xử lý này theo trình tự bất kỳ.

5.3. CÁC THANG DỮ LIỆU TỔNG HỢP

Các biến của bảng dữ liệu phân tích nhận các giá trị trong miền xác định. Các giá trị cho phép của một biến được phản ánh trong việc chọn thang đo. Các hoạt động tổng hợp dữ liệu phụ thuộc vào thang đô, do đó cần phẩn phân loại các tập dữ liệu một cách đầy đủ sao cho dễ nhận biết hoạt động xử lý dữ liệu nào phù hợp với loại thang đo nào. Sau đây là một số loại thang đo phổ biến:

- *Thang đo danh định*: Dữ liệu được đo trên một thang đo danh định nếu dữ liệu có thể đếm được nhưng không xếp hạng được. Ví dụ, sản phẩm, tên nhân viên kinh doanh, nhóm kinh doanh là có thể đo được trên thang đo danh định.
- *Thang đo thứ tự*: Dữ liệu được đo trên thang đó thứ tự nếu dữ liệu có thể đếm được và các giá trị có thể được xếp hạng theo một trình tự có ý nghĩa. Khi một thuộc tính chấp nhận thang đo thứ tự, có nghĩa là chúng ta biết được rằng các giá trị có thể được xếp hạng nhưng sự khác biệt giữa các giá trị là không biết được. Ví dụ về các giá trị có thể đo trong thang đo thứ tự là các lựa chọn trong câu hỏi đa lựa chọn có các đáp ứng là ‘Rất không đồng ý’, ‘Hơi không đồng ý’, ‘Trung lập , ’Hơi đồng ý’, ‘Rất đồng ý’. Trong ví dụ này, có thứ tự xếp hạng đối với câu trả lời cho câu hỏi. Với các thang đo thứ tự, chúng ta biết được rằng một giá trị này lớn hơn giá trị kia nhưng chúng ta không biết là cao hơn bao nhiêu. Sự khác nhau giữa các giá trị là không biết và không thể đo được. Ví dụ, chúng ta không thể nói một cách chính xác ‘Rất đồng ý’ tốt hơn ‘Hơi đồng ý’ bao nhiêu. Và chúng ta không thể đo được liệu sự khác nhau giữa ‘Rất đồng ý’ và ‘Hơi đồng ý’ là lớn hơn hay nhỏ hơn sự khác nhau giữa ‘Hơi đồng ý’ và ‘Trung lập’.
- *Thang đo khoảng*: Đây là thang đo ở đó dữ liệu có thể đếm được, các giá trị có thể xếp hạng được và sự khác biệt bao nhiêu giữa các giá trị là rõ ràng. Ví dụ các thuộc tính có kiểu dữ liệu nhãn thời gian Time Stamp chúng ta biết được chính xác hai ngày cách nhau bao nhiêu ngày.

  Đặc tính xác định của các thang đo khoảng là chúng ta có thể đếm các giá trị, xếp hạng chúng và cộng trừ chúng.
- *Thang tỷ lệ*: Thang đo này giống thang đo khoảng ở mội khía cạnh và thêm vào đó giá trị 0 được xác định phân biệt. Điểm không biểu thị thuộc tính có số lượng bằng không. Các ví dụ về thang đo tỷ lệ là lượng bán, giá sản phẩm. Lượng bán tại điểm không có nghĩa là không có sản phẩm nào được bán. Giá tại điểm không nghĩa là không phải trả tiền cho sản phẩm. Sự tồn tại của điểm không là quan trọng vì nó phân biệt giữa thang tỷ lệ và thang khoảng và nó cho phép chúng ta nhân, chia các giá trị dữ liệu mà không thể thực hiện được với thang khoảng. Các giá trị tỷ lệ cho phép diễn đạt một giá trị là một phần của giá trị khác ở cùng chiều.

  Tóm lại, đặc tính của thang đo tỷ lệ là các giá trị có thể đếm được, xếp hạng được, cộng trừ được và nhân chia được.

Sự phân loại các thang đo tỷ lệ như trên có đặc tính là thang đo sau luôn bao hàm các đặc tính của thang đo trước. Ví dụ, thang đo khoảng bao hàm các đặc tính của thang đo thứ tự và thang đo danh định v.v.

Một số nhầm lẫn thường hay gặp phải khi lần đầu tiên làm việc với các thang đó. Thứ nhất, nhầm lẫn giữa thang đo dang định và thang đo thứ tự. Ví dụ, các nhân viên bán hàng là thang đo danh định vì chúng ta có thể đếm được. Chúng ta có thể yêu cầu trình phân tích SQL sắp xếp các nhân viên theo thứ tự alphabet của tên nên có thể hiểu nhầm các nhân viên bán hàng ở thang đo thứ tự. Tuy nhiên, sự sắp xếp thứ tự các nhân viên kinh doanh là thứ tự ngẫu nhiên theo tên và không phản ánh xếp hạng chất lượng của các nhân viên kinh doanh.

Thứ hai, các giá trị danh định thường bị nhầm lẫn là có thể sắp xếp được bằng cách sử dụng các phép đo khác. Ví dụ, chúng ta có thể sắp xếp các nhân viên kinh doanh theo năng suất bán hàng do đó các nhân viên bán hàng phải là ở thang đo thứ tự chứ không phải ở thang đo danh định? Tuy nhiên, trong ví dụ này chúng ta xếp hạng năng suất bán hàng và ẩn dụ kết hợp hai thuộc tính nhân viên bán hàng và năng suất bán hàng.

Thứ ba, thang đo khoảng thường bị nhầm lẫn với thang đo tỷ lệ ở chỗ sự khác nhau giữa các giá trị khoảng có thể nhân, chia được. Ví dụ, hai lần giai đoạn hai tháng là một giai đoạn bốn tháng do đó sự khách biệt có thể được biểu diễn trong thang đo tỷ lệ.

Cuối cùng, sự khác biệt tinh tế giữa cách chúng ta xếp hạng dữ liệu trên thang đo thứ tự và cách chúng ta sắp xếp thứ tự khi chúng ta lấy dữ liệu như là một câu truy vấn từ bảng quan hệ. Ví dụ, nếu chạy câu lệnh SQL sau:

```
SELECT ten
FROM        trang_thai_khach_hang
ORDER BY    ten
```

Thứ tự của đáp ứng sẽ là theo thứ tự alphabet như sau:

```
ten
==================
Lien_lac
Khach_hang_thuc_su
Khach_hang_tiem_nang
Khach_hang_trien_vong
```

Tập kết quả theo trình tự không phải là cái chúng ta mong muốn trong thang thứ tự. Trình phân tích cú pháp SQL không biết cách xếp hạng các thực thể theo thang đo thứ tự theo voàng đời của khách hàng. Giải pháp là tạo thêm một thuộc tính ‘hang’ cho phép chúng ta sắp xếp các hàng theo cách mong muốn. Ví dụ, câu lệnh SQL sau:

```
SELECT      hang, ten
FROM        trang_thai_khach_hang
ORDER BY    hang
```

Sẽ cho kết quả sau:

```
Hang        ten
=====       ==================
1           Lien_lac
2           Khach_hang_tiem_nang
3           Khach_hang_trien_vong
4           Khach_hang_thuc_su
```

5.4. CÁC TÙY CHỌN TỔNG HỢP DỮ LIỆU

Trong phần này chúng ta tìm hiểu cách tính toán các giá trị tổng hợp. Một giá trị tổng hợp là một giá trị biểu diễn một nhóm các giá trị của cùng một thuộc tính nhưng từ các thể hiện (hàng) khác nhau. Ví dụ, chúng ta có các giá trị 3,5, và 7 là giá của ba sản phẩm, giá trị tổng hợp có thể là 15 biểu diễn tổng giá trị các sản phẩm. Giá trị tổng hợp khác có thể cho ví dụ này là 5 biểu diễn giá trị giá trung bình của ba sản phẩm. Sau đây là các hoạt động tổng hợp dữ liệu thông dụng cho các biến. Như đề cập ở phần trước, thang đo của biến là yếu tố quyết định trong lựa chọn tổng hợp dữ liệu.

Đếm (count): Cũng được biến đến như là tần suất. Phép đo này biểu diễn số lần xảy ra của các giá trị dữ liệu cụ thể trong tập dữ liệu. Tần suất có thể được sử dụng với tất cả các kiểu biến và dùng được với các biến đo danh định.

Tối thiểu, tối đa: Cung cấp giá trị thấp nhất và cao nhất trong hạng. Các biến cần phải được điều chỉnh tỷ lệ cho một hạng để có nghĩa do đó các biến đo đanh định không áp dụng được. Các biến ở thang đo còn lại có thể sử dụng lựa chọn này để tổng hợp dữ liệu.

Tính tổng: Đây là góa trị tổng khi tất cả các giá trị dữ liệu được tổng lại. Tất nhiên, hoạt động tổng hợp này chỉ có ý nghĩa khi các giá trị dữ liệu có thể được tính toán số học bao gồm tối thiểu là phép cộng trừ các giá trị. Do đó, hoạt động tính tổng cho tổng hợp dữ liệu chỉ áp dụng cho các biến đo khoảng và thang đo tỷ lệ.

Tính trung bình: Các phép đo trung bình cho phép tổng hợp thông tin về tâm của phân bố các giá trị dữ liệu. Có các phép đo trung bình khác nhau tùy thuộc vào thang đo. Với các biến đo danh định, phép đo tổng hợp phù hợp là *mode*. Mode là giá trị xảy ra nhiều nhất. Với các biến đo thứ tự, phép đo tổng hợp trung bình là *trung vị*. Với các biến đo khoảng và tỷ lệ, phép đo tổng hợp trung bình là giá trị trung bình. Giá trị trung bình biểu diễn trung bình số học của các giá trị.

Biến động (variation): Các phép đo biến động cung cấp thông tin về sự dàn trải của dữ liệu xung quanh tâm. Phép đo này không áp dụng cho các biến đo danh định do không tồn tại tâm. Với thang đo thứ tự, khoảng tứ phân vị (IRQ) có thể được xác định. Đây là khoảng các giá trị đại diện nửa giữa của các giá trị. Với các biên đo khoảng và tỷ lệ, phép đo biến động là độ lệch chuẩn (SD). Độ lệch chuẩn đo dữ liệu gần hay xa so với giá trị trung bình.

Dạng (shape):Phép đo tổng hợp về dạng mô tả một cách chi tiết sự dàn trải của dữ liệu quanh tâm. Kurtosis và độ xoắn là các phép đo phổ biến nhất cho các biến đo khoảng và tỷ lệ. Kurtosis biểu thị độ phẳng của phân bố dữ liệu: kurtosis càng cao thì dữ liệu càng gần với giá trị trung bình và ngược lại. Độ xoắn biểu thị tính đối xứng của dạng. Phép đo độ xoắn xác định có hay không dữ liệu tập trung hơn ở một phía của giá trị trung bình so với phía khác. Các phép đo tổng hợp về dạng thường ít được sử dụng hơn so với các phép đo khác.

5.5. BẢNG TÓM TẮT VÀ BẢNG TẦN SỐ

Một khái niệm quan trọng trong tổng hợp dữ liệu là các thuộc tính *dẫn xuất (derived attibute)*. Thuộc tính dẫn xuất là một thuộc tính mà các giá trị của nó là kết hợp theo một cách định nghĩa trước của các giá trị khác của cùng thể hiện. Ví dụ, thuộc tính 'Lợi tức' có thể được tính bằng cách nhân giá trị của thuộc tính 'Giá' với thuộc tính 'Số lượng'. Thuộc tính dẫn xuất không được lưu trong tập dữ liệu nhưng do biết cách tính chúng nên có thể tính toán ra chúng khi cần.

Giá trị dẫn xuất luôn tham chiếu đến một thể hiện. Do đó, nó khác với giá trị tổng hợp ở đó luôn tham chiếu đến nhiêu thể hiện dữ liệu. Bảng 5.4 minh họa sự khác biệt giữa giá trị dẫn xuất và giá trị tổng hợp. Ở cột cuối cùng của bảng ('Loi_tuc'), các giá trị được viết chữ in nghiêng là các giá trị dẫn xuất còn giá trị được viết chữ in thường là giá trị tổng hợp.

Bảng 5.4 Ví dụ về sự khác biệt giữa giá trị dẫn xuất và giá trị tổng hợp.

Don_hang	San_pham	Gia	So_luong	Loi_tuc
1	Cam	1	6	*6*
1	Nến	7	34	*238*
				244

Bảng tính là công cụ mạnh để tạo các giá trị dẫn suất sử dụng các công thức đã được định nghĩa. Các công thức có thể tham chiếu đến các ô và có thể được sao chép và dán một cách linh hoạt.

Một bảng tóm tắt cung cấp dữ liệu tổng hợp của một thuộc tính được nhóm lại theo các giá trị của một thuộc tính khác. Bảng tóm tắt cung cấp một tập con các cột và hàng và được coi là khung nhìn xúc tích của tập kết quả giải chuẩn. Các gói phần mềm trí tuệ doanh nghiệp cho phép tóm tắt và đào sâu dữ liệu một cách tương tác. Bảng 5.5 minh họa bảng tóm tắt cho biến đo danh định ở đó tổng lợi tức được tóm tắt theo nhóm bán hàng

Bảng 5.5 Bảng tóm tắt cho biến danh định -Tổng lợi tức chia theo nhóm bán hàng.

Nhom_bh	Loi_tuc
Alpha	*1,051*

Beta	*889*
	1,940

Các bảng tóm tắt có thể được tạo cho các biên khoảng như là nhãn thời gian. Ví dụ, số thứ tự tuần có thể được sử dụng để tách và tổng hợp dữ liệu như ở bảng 5.6.

Bảng 5.6 Bảng tóm tắt cho biến khoảng -Tổng lợi tức chia theo tuần.

Tuan	Loi_tuc
1	*244*
2	*692*
3	*260*
4	*424*
4	*324*
	1,940

Một phiên bản đặc biệt của bảng tóm tắt là bảng tần số. Trong một bảng tần số, các giá trị của một thuộc tính được nhóm vào một cột và cột thứ hai chỉ thị số lần xảy ra của mỗi giá trị trong tập kết quả gốc. Bảng 5.7 minh họa một ví dụ cho các biến trong thang đo danh định, trong ví dụ này là sản phẩm.

Bảng 5.7 Đếm số sản phẩm được bán theo sản phẩm được bán

San_pham	Dem
Cam	4
Táo	4
Rượu	4
Xà phòng	4
Nến	4
	20

Các bảng tần suất có thể được xây dựng cho các biến khoảng và tỷ lệ. Trong trường hợp đó, các ngăn sẽ được xác định. Các ngăn được cắt các phần của thang đo sao cho chúng biểu diễn toàn bộ khoảng ở đó các biến được đo. Mỗi ngăn biểu diễn một phần bằng nhau của thang đo.

Khi các ngăn được xây dựng, các giá trị dữ liệu nằm trong một ngăn có thể được đếm số lần xuất hiện.

5.6. BẢNG CROSS-TAB VÀ BẢNG PIVOT

Một bảng cross-tab là một bảng tóm tắt cho hai chiều. Giá trị của bảng cross-tab cung cấp cái nhìn sâu sắc trong mối quan hệ giữa hai biến trong đó cross-tab chia nhỏ số tổng. Ví dụ, bảng 5.8 cung cấp thông tin tổng quan về lợi tức kinh doanh của mỗi nhóm bán hàng mỗi tháng.

Bảng 5.8 Bảng cross-tab lợi tức chia theo nhân viên kinh doanh theo tuần

Nhan_vien	Nhom_bh	Tuan_1	Tuan_2	Tuan_3	Tuan_4	Tuan_5	Tong
Nam	Alpha	244	75	0	0	0	319
Hương	Alpha	0	279	29	424	0	732
Tuấn	Beta	0	108	231	0	294	633
Thủy	Beta	0	230	0	0	26	256
		244	692	260	424	320	1,940

Bảng 5.8 cho chúng ta cái nhìn về các hoạt động của nhân viên kinh doanh theo các tuần. Chúng ta có thể thấy rằng nhân viên tên Nam ở nhóm Alpha rất thành công ở tuần 1 và tuần 2 và không thành công ở các tuần sau đó. Thông tin này không sẵn sàng để có thấy được trong bảng dữ liệu gốc. Giống như thế, ở bảng 5.9 có thể thấy rằng các đơn hàng lớn chỉ có với rượu và nến.

Bảng 5.9 Bảng cross-tab của sản phẩm được bán chia theo lợi tức theo sản phẩm được bán

Loi_tuc	Cam	Táo	Rượu	Xà phòng	Nến	Tong
0-50	4	4	0	1	1	10
51-100	0	0	0	3	0	3
101-150	0	0	0	0	0	0
151-200	0	0	3	0	0	3
201-250	0	0	1	0	3	4
	4	4	4	4	4	120

Các hàng và cột trong bảng cross-tab có ý nghĩa khác với các hàng và cột trong bảng quan hệ hoặc tập kết quả SQL. Trong bảng quan hệ, các cột được co là các thuộc tính của các thực thể và các hàng là các thể hiện của các thực thể. Trong bảng cross-tab, cả hàng và cột là các giá trị tổng hợp.

Một bảng pivot là phiên bản tương tác của bảng cross-tab. Bảng pivot cho phép tạo các cross-tab một cách tương tác. Để tạo bảng pivot, băt đầu với các bảng dữ liệu gốc. Sử dụng bảng pivot, các cấu trúc tóm tắt có thể được thay đổi bằng cách đổi chỗ các hàng và các cột mà chúng ta thấy phù hợp sử dụng phương pháp kéo – nhả tương tác.

CHƯƠNG 6: HIỂN THỊ DỮ LIỆU

- *Chọn biểu đồ phù hợp cho loại biến phù hợp hoặc tập các biến*
- *Hiểu mối nguy hiểm của đề xuất phụ thuộc khi sự phụ thuộc không tồn tại*
- *Thảo luận ưu nhược điểm của thêm màu vào hiển thị dữ liệu*
- *Hạn chế trang trí biểu đồ với các hiệu ứng*

6.1. TẦM QUAN TRỌNG CỦA VIỆC HIỂN THỊ DỮ LIỆU

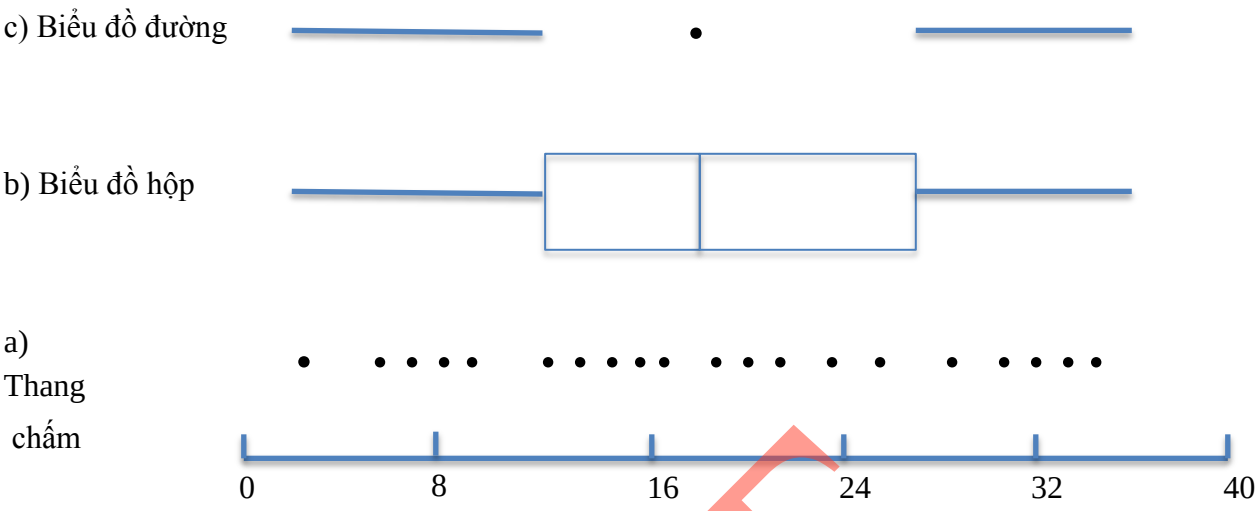
Trong chương 5, chúng ta đã tìm hiểu việc xây dựng các bảng tổng hợp như bảng tần số, bảng tóm tắt, bảng cross-tab và bảng pivot. Trong chương này, chúng ta sẽ tìm hiểu phương pháp hiện thị trực quan các bảng này. Dù rằng các biểu đồ thường sử dụng các dữ liệu tổng hợp làm cơ sở, nhứng các dữ liệu giao dịch thô cũng có thể được hiển thị trực quan mà không cần hoạt động tổng hợp. Việc hiển thị dữ liệu giúp khám phá các mẫu trong dữ liệu và giúp hiểu các khuynh hướng không dễ thấy ở các bảng tổng hợp. Việc hiện thị dữ liệu một cách hợp lý cũng giúp tránh quá tải thông tin đối với người phân tích dữ liệu. Sự ưu việt của việc sử dụng các biểu đồ trong phân tích dữ liệu so với các bảng tổng hợp đã được chứng minh bằng thực nghiệm người dùng phân tích dữ liệu và ra quyết định.

Có sự bất đối xứng thú vị giữa dữ liệu được trình bày ở dạng bảng so với dữ liệu được trình bày ở dạng đồ họa. Dữ liệu dạng bảng luôn có thể được chuyển đổi thành dạng đồ thị nhưng điều ngược lại thị không luôn đúng mà không gây tổn thất về độ chính xác. Có thể nói, sự đánh đổi khi sử dụng biểu diễn đồ họa so với biểu diễn dạng bảng là có lợi về hiểu biết dữ liệu nhưng mất về độ chính xác của dữ liệu. Khi cả hai yếu tố đều cần thì phải dùng cả hai dạng biểu diễn đồ thị và bảng.

Các biểu đồ có thể được phân loại theo số các thuộc tính được trình bày cùng lúc. Chương này sẽ trình bày về các biểu đồ hiển thị các giá trị trên: 1)một biến, 2) hai biến với các bảng tóm tắt và bảng cross-tab và 3) ba hoặc nhiều biến.

6.2. HIỂN THỊ MỘT BIẾN

Loại biểu đồ đơn giản nhất là các biểu đồ đơn biến. Các biểu đồ này hiển thị một tập các giá trị của một biến cụ thể. Trong các trường hợp các biến được đo trong các thang đo thứ tự, khoảng và tỷ lệ, các thang đo này đủ thông tin để có thể được vẽ các giá trị dữ liệu theo một trục. Các dữ liệu riêng biệt và tổng hợp có thể được sử dụng để vẽ các biến trên các trục. Một trục biểu diễn một chiều của dữ liệu. Hình 6.1 minh họa ví dụ về các biểu đồ vẽ các biến thang thứ tự, khoảng và tỷ lệ trên một trục.

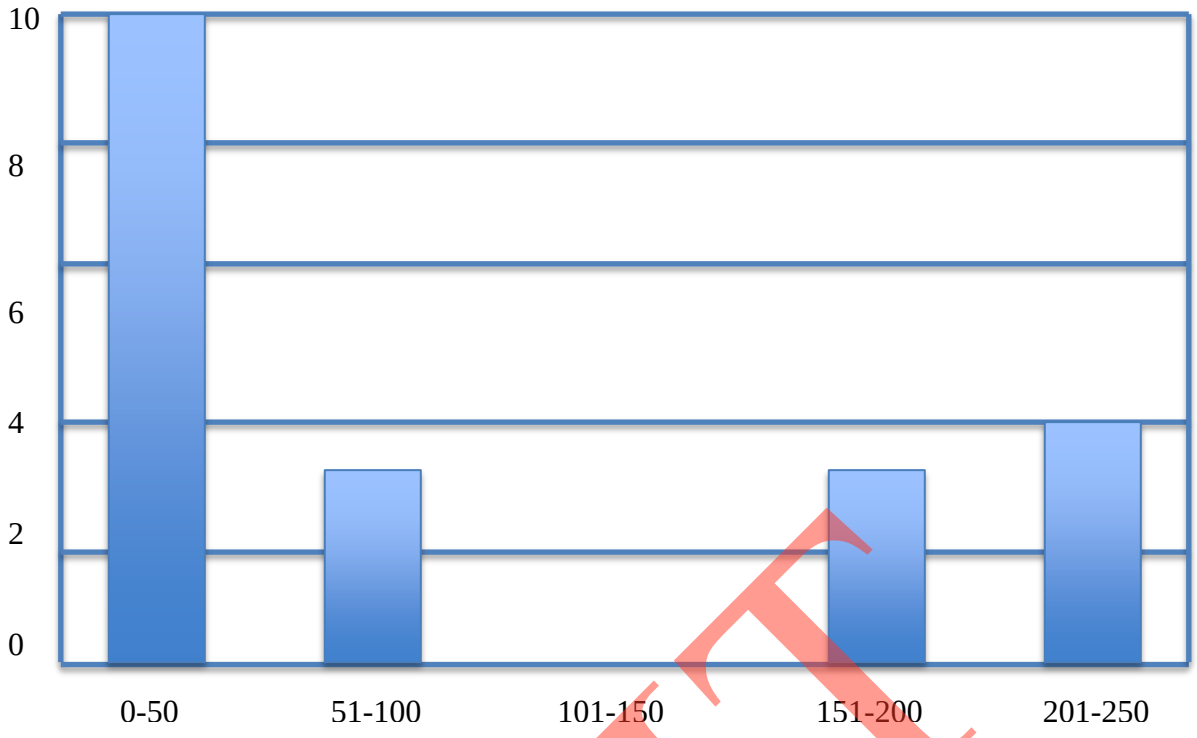


Hình 6.1: Thang chấm, biểu đồ hộp và biểu đồ đường.

Hình 6.1 (a) biểu diễn thang chấm ở đó mỗi giá trị được biểu diễn bởi một chấm trên thang của biến. Biểu diễn này cho ta hiểu biết về phân bố của các giá trị dữ liệu riêng biệt trên toàn dải giá trị.

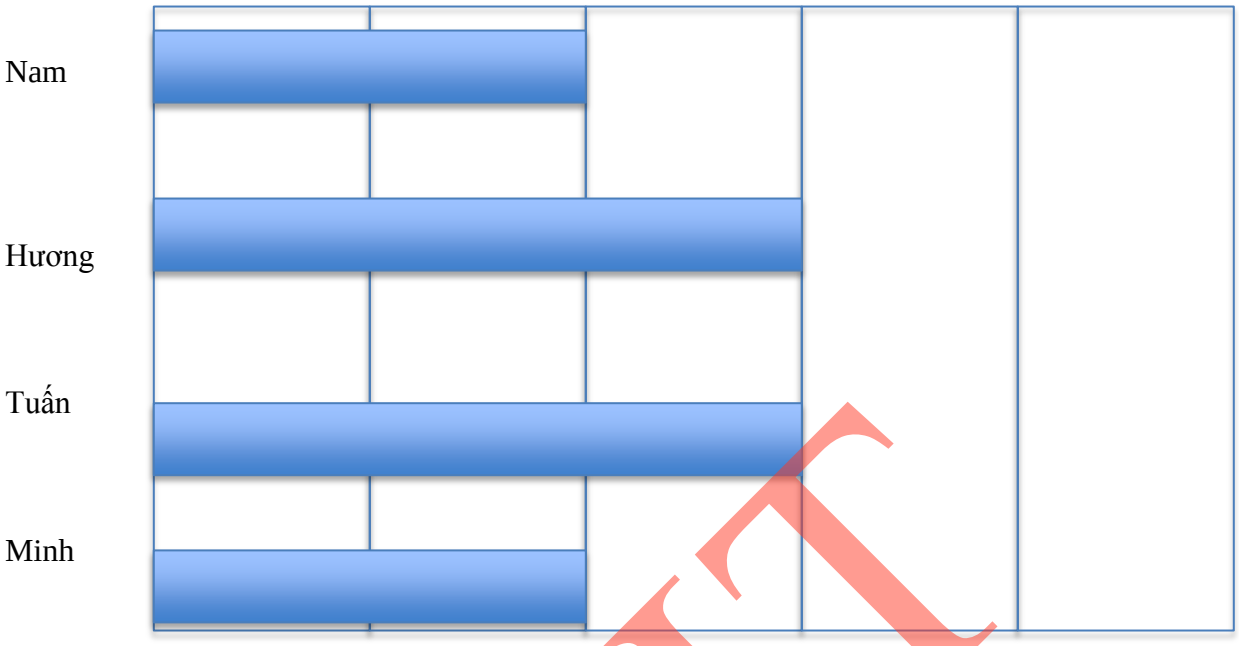
Thang chấm bắt đầu mất giá trị của nó khi có nhiều điểm dữ liệu và phân bố các giá trị dọc trục trở lên không rõ ràng. Trong trường hợp này, biểu diễn các phép đo tổng hợp thống kê sẽ hiệu quả hơn. Với các biến thứ tự, khoảng và tỷ lệ, chúng ta có thể sử dụng giá trị tối đa, giá trị tối thiểu, tứ phân vị thứ nhất, trung vị và thứ phân vị thử 3 để vẽ sử dụng biểu đồ hộp. Trong biểu đồ hộp, hộp bắt đầu ở từ phân vị thứ nhất và kết thúc ở tứ phân vị thứ ba. Dải này trong hộp là dải liên tứ phân vị (một nửa cá giá trị). Đường thẳng bên trong hộp[biểu diễn giá trị trung vị. Các đường bên ngoài hộp gọi là các viền phân nhánh tới giá trị thấp nhất và cao nhất. Biểu đồ đường Hình 1 (c) biểu diễn phép đo thống kê giống như biểu đồ hộp chỉ khách ở chỗ giá trị trung vị được biểu diễn bởi một dấu chấm. Khoảng trắng giữa các đường và dấu chấm biểu diễn dải liên tứ phân vị.

Một dạng hiện thị tiếp theo là hiển thị các bảng tần số. Bảng tần số là một dạng bảng tóm tắt đặc biệt cung cấp các phép đo tổng hợp cho một biến. Loại biểu đồ được chọn để hiển thị các bảng tần số phụ thuộc vào thang đo của biến. Nếu có một bảng tần số cho các biến thang thứ tự, khoảng hoặc tỷ lệ, biểu đồ histogram thường được sử dụng. Hình 6.2 minh họa ví dụ biểu đồ histogram biểu diễn tổng số các sản phẩm trên các dải giá trị lợi tức. Một biểu đồ histogram, trục hoành biểu diễn các khoảng giá trị và trục tung biểu diễn tần số.



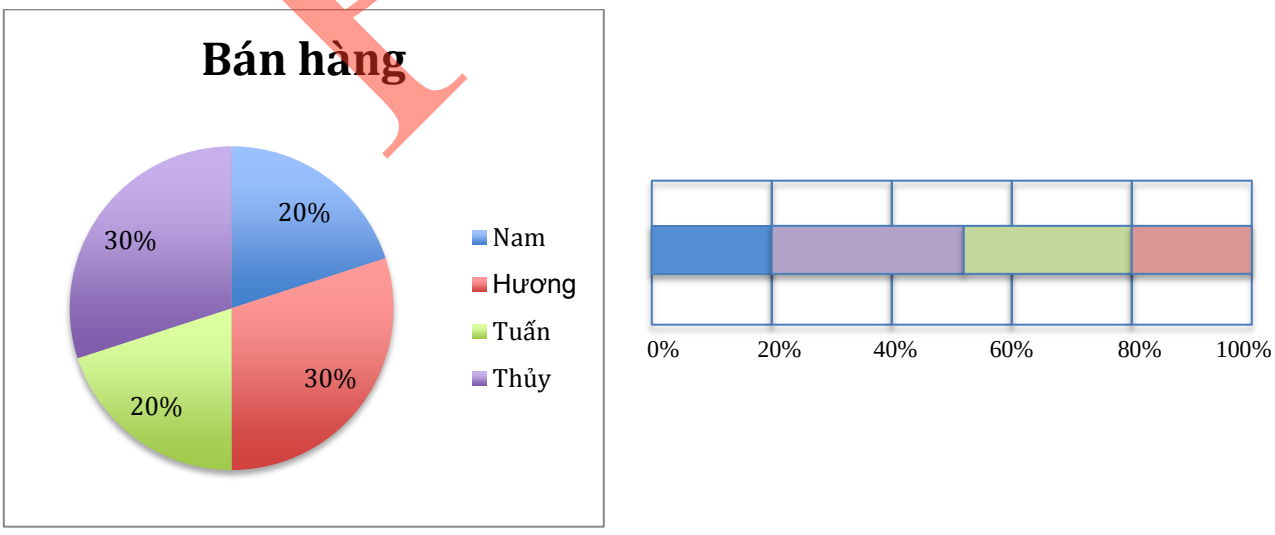
Hình 6.2 Biểu đồ histogram

Nếu chúng ta biểu diễn đồ thị một biến thang danh định, biểu đồ thanh (bar chart) có thể được sử dụng để biểu diễn bảng tần số. Một biểu đồ thanh giống như biểu đồ histogram nhưng các giá trị danh định không biểu diễn trình tự nào cả. Để tránh nhầm lẫn với biểu đồ histogram, biểu đồ thanh thường được miêu tả theo chiều ngang. Hình 6.3 là một ví dụ minh họa về sử dụng biểu đồ thanh biểu diễn số lần mỗi nhân viên kinh doanh thực hiện thành công một đơn hàng.



Hình 6.3: Biểu đồ thanh.

Nếu chúng ta quan tâm đến tần số tương đối hơn là tần số tuyệt đối, một biến có thể được biểu đồ hóa với các biểu đồ dạng tròn (pie chart) hoặc biểu đồ thanh chồng xếp (stacked bar chart). Loại biểu đồ này được nhấn mạnh biểu diễn các tỷ lệ phần trăm các giá trị dữ liệu. Hình 6.4 là một ví dụ về biểu đồ dạng tròn và biểu đồ thanh chồng xếp.

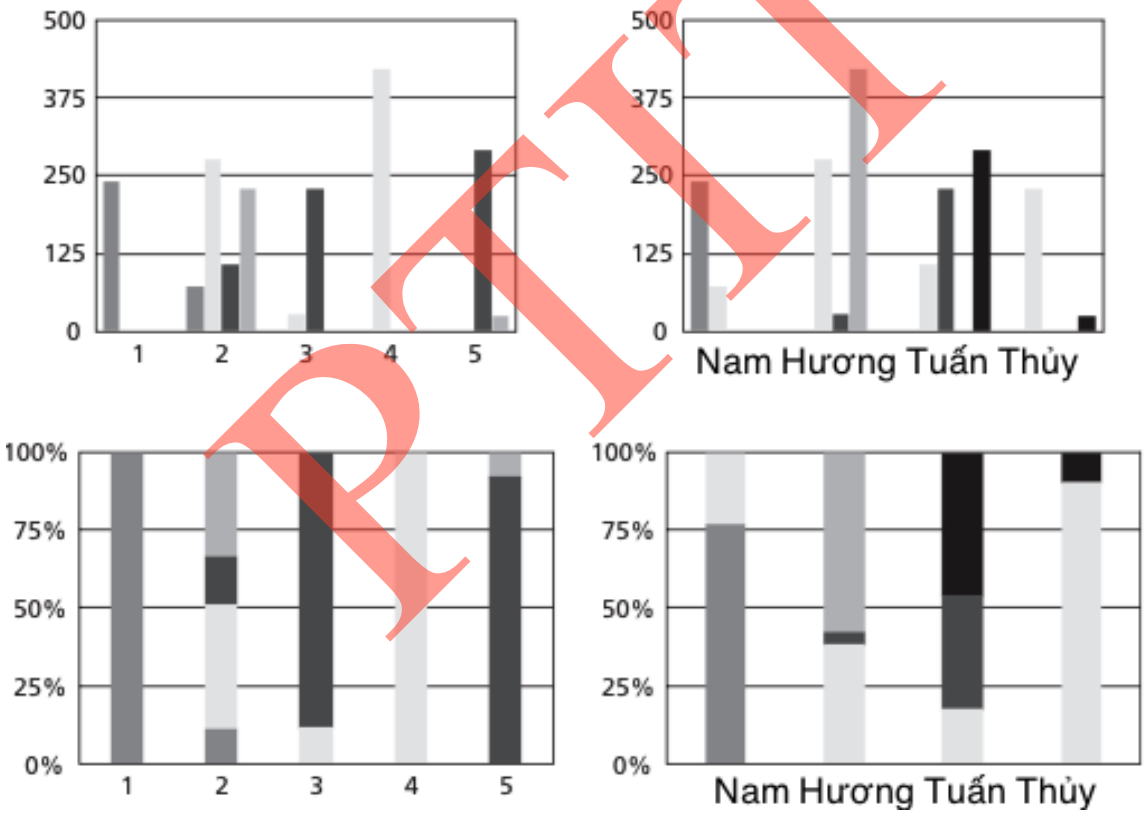


Hình 6.4: Biểu đồ dạng tròn và biểu đồ thanh chồng xếp.

6.3. HIỂN THỊ HAI BIẾN

Các bảng tóm tắt có thể cung cấp dữ liệu tổng hợp của một biến được nhóm theo các giá trị của biến khác. Trong trường hợp này, các dữ liệu tổng hợp thường biểu diễn ở thang đo khoảng hoặc tỷ lệ và các biến nhóm thường ở thang danh định. Bảng tổng hợp giống như bảng tần số có thể được biểu đồ hóa sử dụng các biểu đồ thanh và biểu đồ dạng tròn. Như đã biết, các biểu đồ thanh nhấn mạnh biểu diễn các tần số tuyệt đối còn các biểu đồ dạng tròn và biểu đồ thanh chồng xếp nhấn mạnh bểu diễn các tấn số tương đối.

Các bảng cross-tab và bảng pivot là các bảng tóm tắt được tách làm ma trận hai nhân hai. Do đó, chúng biểu diễn cùng các giá trị tổng hợp được chia làm hai chiều. Cách thông thường để biểu đồ hóa các bảng cross-tab sử dụng các biến thể đặt cạnh nhau ((side-by-side) của biểu đồ đơn biến. Hinh 6.5 minh họa cách cách khác nhau sử dụng các biến thẻ của biểu đồ đơn biến để biểu đồ hóa bảng cross-tab. Dữ liệu nguồn cho các biểu đồ này lấy từ Bảng 5.8.



Hình 6.5: Biểu đồ hóa dữ liệu bảng tóm tắt cross-tab (dữ liệu nguồn từ Bảng 5.8).

Biểu đồ góc trên bên trái Hình 6.5 đối chiếu các nhân viên kinh doanh với nhau và sắp xếp thứ tự theo tuần. Biểu đồ góc trên bên phải đối chiếu các tuần và sắp xếp theo nhân viên kinh doanh. Các hai biểu đồ tập trung vào các giá trị tuyệt đối.

Hai biểu đồ bên dưới trong hình 6.5 là các biến thể side-by-side của biểu đồ thành chóng xếp tập trung và các tấn số tương đối các giá trị dữ liệu.

Các biến thể side-by-side của các biểu đồ đơn biến ngược với các biểu đồ nhị biến biểu diễn mối quan hệ giữa một biến và một biến khác. Tên kỹ thuật cho biểu diễn là hiệp biến vì chúng ta đang cố xác định hai biết có thay đổi cùng nhau hay không. Nói một cách khác, liệu thay đổi của một biến có kéo theo thay đổi biến khác hay không. Các biểu đồ nhị biến thương được sử dụng cho các biến thang khoảng và thang tỷ lệ,

Các biểu đồ nhị biến ánh xạ các thang tỷ lệ của biến thứ nhất và biến thứ hai lên trục hoành và trục tung. Việc tỷ lệ hóa không bị biến dạng có nghĩa là một dải trên một trục biểu diễn dải tương đương trên thang tỷ lệ.

Khi dây dựng các biểu đồ nhị biến, chúng ta có thể khai báo các biến như là biến độc lập và các thuộc tính khác là các biến phụ thuộc. Biến độc lập là biến biểu diễn nguyên nhân. Biến phụ thuộc biểu diễn kết quả. Do đó chúng ta đã định nghĩa chiều hiệp biến. Nếu biến A là biến độc lập và biến B là biến phụ thuộc nghĩa là thay đổi biến A sẽ dẫn đến thay đổi biến B nhưng không đúng theo chiều ngược lại. Việc khai báo phụ thuộc giữa các biến đóng vai trò quan trọng trong biểu diễn biểu đồ vì biến phụ thuộc sẽ được biểu diễn trên trục hoành còn biến phụ thuộc sẽ được biểu diễn trên trục tung.. Việc khai báo sự phụ thuộc giữa các biến không có nghĩa là các biến đó thực sự có mối quan hệ phụ thuộc. Tribg một số trường hợp, các dữ liệu nhìn có vẻ có mối quan hệ phụ thuộc giữa hai biến nhưng trong thực tế không tồn tại mối quan hệ phụ thuộc.

Để minh họa việc đồ thị hóa các biểu đồ nhị biến, xét tập dữ liệu biểu diễn trong bảng 6.1. Tập dữ liệu biểu diễn 4 biến và hai mươi thể hiện. Chúng ta sẽ tập trung vào biến Var1 và Var2.

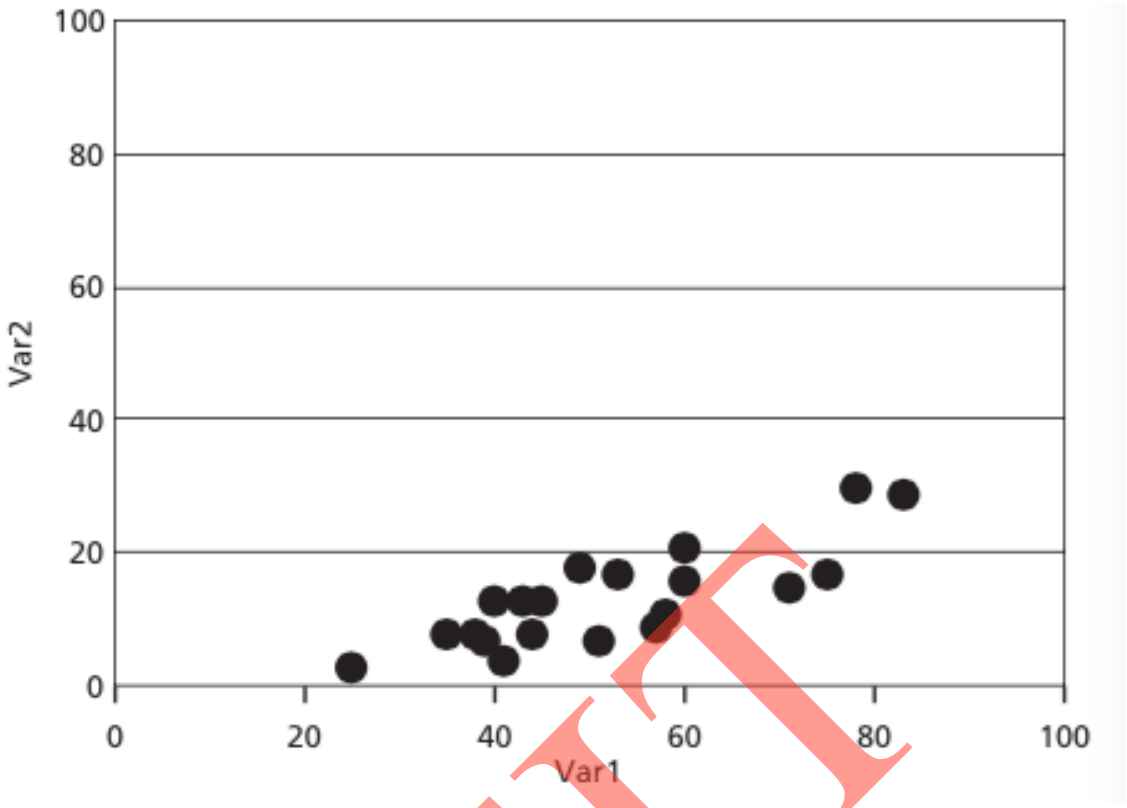
Biểu đồ phân phấn là mở rộng nhị biến của biểu đồ chấm ở đó mỗi chấm biểu diễn hai giá trị, một trên trục hoành và một trên trục tung. Chú ý rằng việc biểu đồ hóa một biểu đồ phân tán không hàm ý là các biến hiệp biến hoặc tồn tại sự phụ thuộc giữa chúng.

Bảng 6.1: Dữ liệu đa biến mẫu.

Entity	Var1	Var2	Var3	Var4
1	60	21	40	8
2	35	8	33	5
3	78	30	60	14
4	71	15	51	10
5	53	17	43	10
6	57	9	35	6
7	25	3	25	7
8	45	13	38	11
9	49	18	43	13
10	43	13	29	8
11	39	7	32	7
12	51	7	35	5
13	44	8	35	5
14	75	17	54	18
15	38	8	46	9
16	83	29	49	13
17	60	16	52	12
18	41	4	27	5
19	40	13	35	7
20	58	11	47	8

Mục đích của biểu đồ phân bố là để tìm xem có tồn tại hiệp biến giữa hai biến hay không. Hình 6.6 vẽ biểu đồ phân bố cho hai biến Var1 và Var2. Có thể thấy từ biểu đồ hình 6.6 sự hiệp biến tuyến tính giữa hai biến Var1 và Var2.

Khi hai biến hiệp biếu, có thể tồn tài sự phụ thuộc giữa các biến. Nhưng vẫn còn có khả năng khác. Thứ nhất, sự phụ thuộc có thể được dự đoán nhứng không tồn tại. Ở đó, sự thay đổi xảy ra ở một biến không thất sự dẫn đến thay đổi ở một biến khác mà là kết quả của trùng hợp ngẫu nhiên. Thứ hai, chiều của phụ thuộc có thể ngược lại. Khả năng thứ ba, các biến thay đổi cùng nhau có thể là kết quả của thay đổi của một biến thứ ba gọi là biến giả và biến giả này có thể không được mô hình hóa và có thể rất khó xác định.

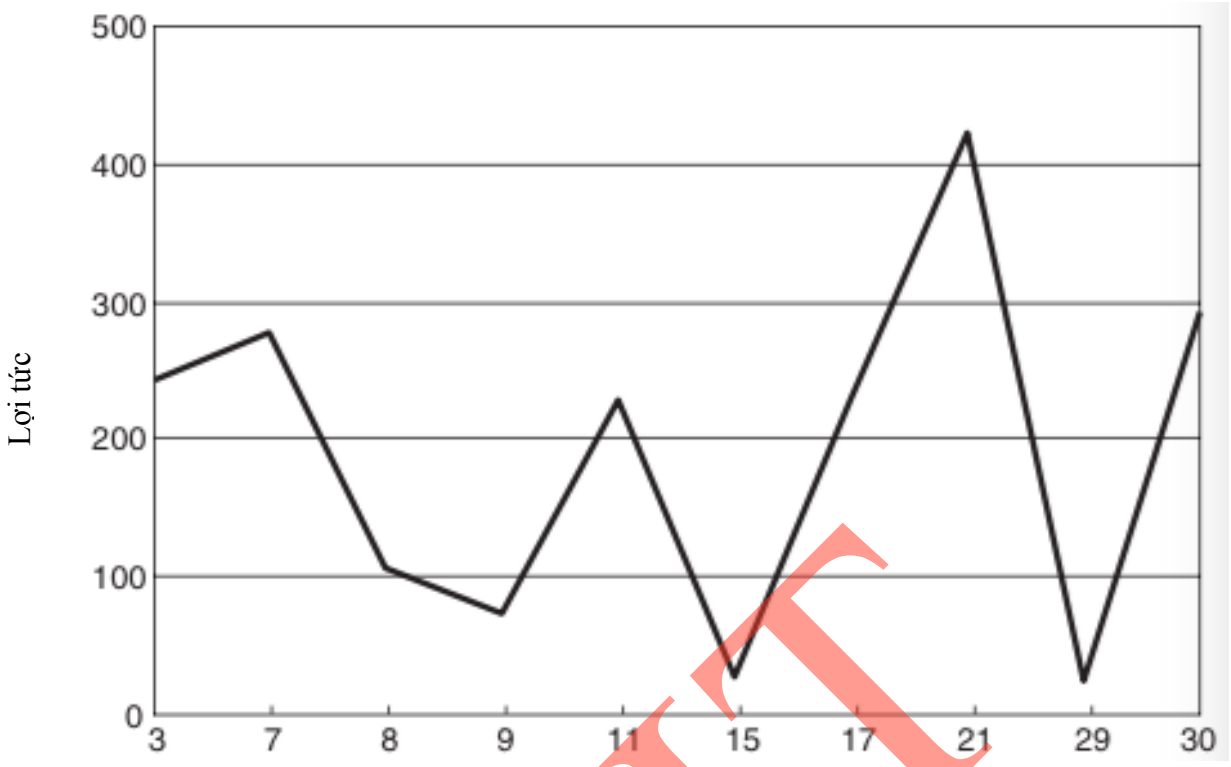


Hình 6.6: Biểu đồ phân bố.

Tóm lại, nếu nghiên cứu mối quan hệ giữa hai biến chúng ta cần cẩn trọng trong việc kết luận mối quan hệ phụ thuộc giữa chúng. Cần suy xét cẩn trọng các khả năng khác. Nếu không chắc chắn, các thang đo của biến có thể hữu dụng trong phân tích phụ thuộc. Dữ liệu danh định thường (không phải luôn luôn) độc lập. Các biến chuỗi thời gian thường (không phải luôn luôn) độc lập.

Loại biểu đồ cuối cùng chúng ta xem xét ở phần này là biểu đồ đường. Biểu đồ đường là một loại đặc biệt của biểu đồ phân bố ở đó một đường được vẽ nối giữa các điểm. Đường gợi ý xu hướng của biến và hàm ý rằng các giá trị trên đường có thể được suy ra.

Một loại biểu đồ đường là biểu đồ chuỗi thời gian sử dụng một biến chuỗi thời gian được hiển thị trên trục hoành và một số biến khác hiển thị trên trục tung. Hình 6.7 laf một biểu đồ đường biểu diễn lthay đổi ợi tức theo thời gian.

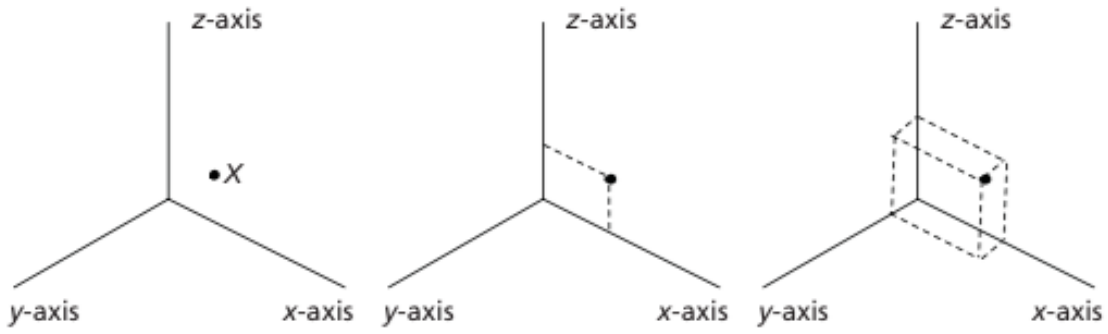


6.3.1.

Hình 6.7: Biểu đồ đường.

6.4. HIỂN THỊ BA HOẶC NHIỀU BIẾN

Có nhiều lựa chọn khi cần phải biểu diễn trực quan ba biến cùng lúc. Lựa chọn thứ nhất, sử dụng các biểu đồ phân bố 3 chiều và biểu đồ đường 3 chiều (trục X, Y, Z). Khi biểu diễn trong không gian 3 chiều, cần có các đường hỗ trợ để tránh nhầm lẫn thị giác về vị trí của các điểm. Hình 6.8 minh họa một ví dụ về nhầm lẫn thị giác về vị trí của một điểm trong không gian.

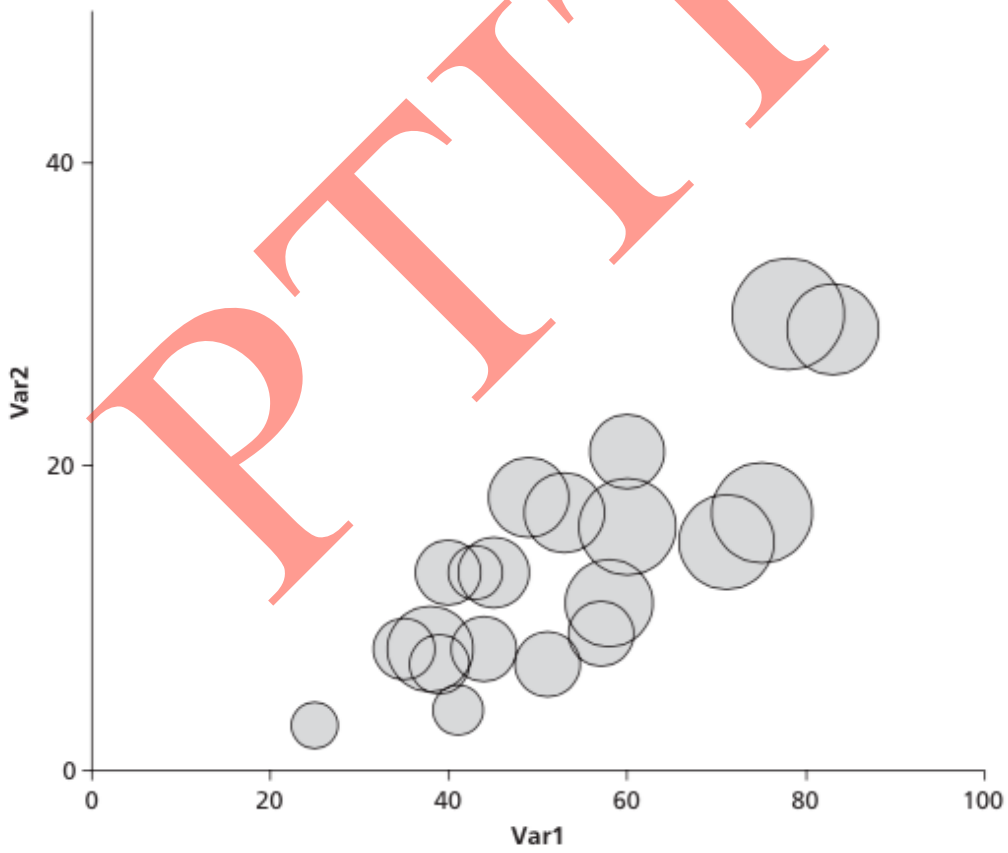


Hình 6.8: Ví dụ về nhầm lẫn thị giác về vị trí trên biểu đồ phân bố 3 chiều.

Giả sử ta muốn biểu diễn các biến Var1, Var2 và Var3 được miêu tả trong Bảng 6.1. Nếu biểu diễn ba biến này sử dụng biểu đồ đường 3 chiều có thể gây ra các nhầm lẫn thị giác. Một lựa chọn khác là sử dụng biểu đồ bong bóng (bubble chart). Một biểu đồ bong bóng giống như biểu đồ phân bố nhị biến chỉ khác ở chỗ chiều thứ 3 được biểu diễn bởi đường kính của chấm. Hình 10 minh họa một ví dụ biểu đồ bong bóng.

Khi xem xét phân tích các mẫu trong biểu đồ bong bóng ở Hình 10, chúng ta cần tìm sự hiệp biến. Quan sát cho thấy các bong bóng có kích cớ lớn gắn với các giá trị lớn của biến Var1 và Var2. Do đó, có vẻ như tồn tại hiệp biến dương ở cả ba chiều.

Một lựa chọn khách cho biểu đồ hóa dữ liệu 3 chiều thường được sử dụng là vẽ rã tất cả các mối quan hệ hai chiều trong ma trận biểu đồ phân tán. Nếu chúng ta muốn biểu diễn trực quan bằng biểu đồ cả ba chiều, cần 3x3x1 = 6 biểu đồ hai chiều. Hình 6.10 minh họa một ma trận biểu đồ hai chiều. Chúng ta có thể áp dụng phương pháp này để biểu diễn dữ liệu nhiều chiều hơn, khi đó số các biểu đồ phân bố hai chiều sẽ tăng theo hàm mũ.

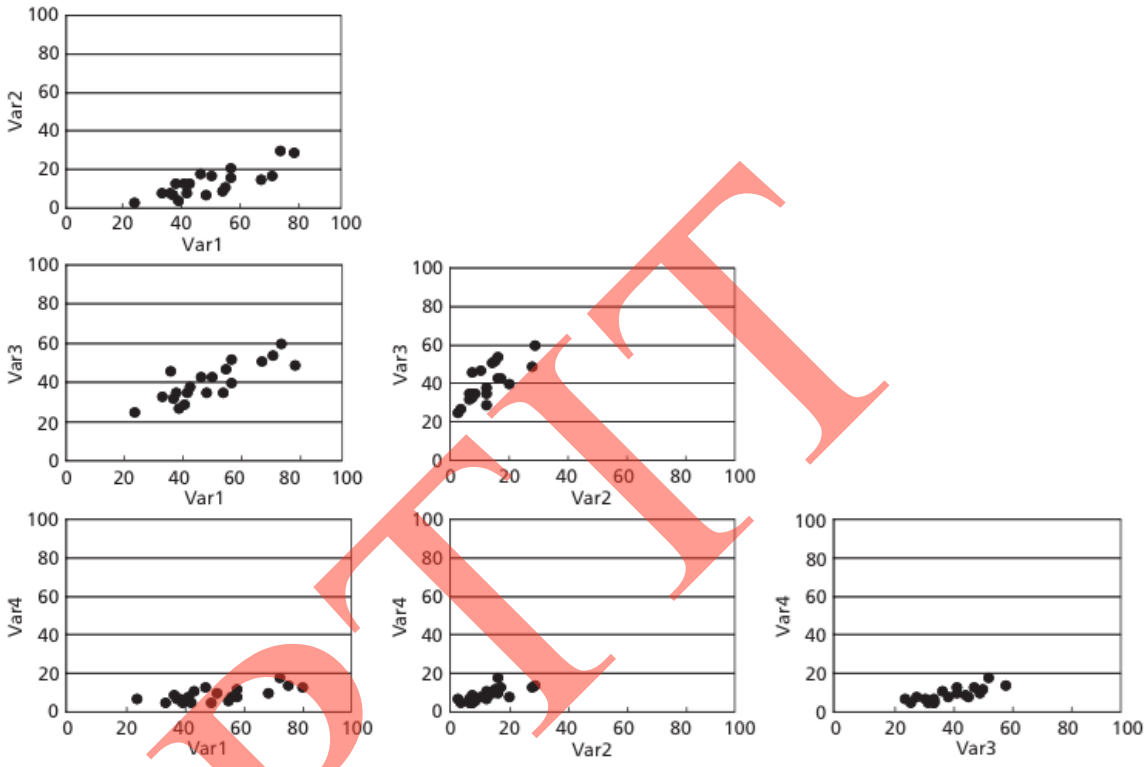


Hình 6.9:Biểu đồ bong bóng.

Để biểu diễn bốn chiều, biểu đồ bong bóng có thể được sử dụng ở đó chiều thứ 4 được biểu diễn bởi khuynh độ mầu và mầu của các bong bóng trong biểu đồ bong bóng với giá trị tương

ứng biểu diễn biến thứ tư. Vấn đề đối với biểu đồ bong bóng tô mầu là rất khó thấy các mối quan hệ giữa các chiều liên quan.

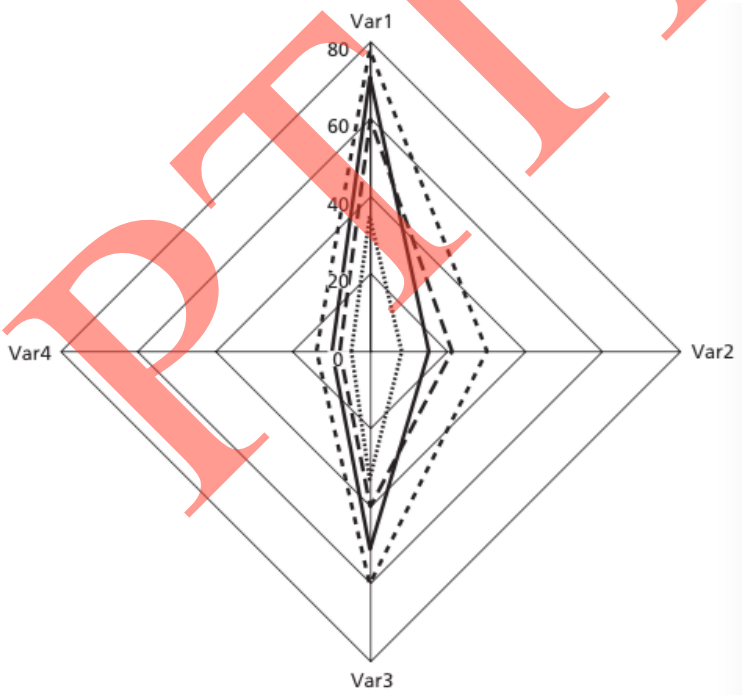
Dữ liệu đa chiều với năm hay sáu chiều hoặc nhiều chiều hơn rất khó biểu diễn trực quan. Một số kỹ thuật cần được áp dụng để biểu diễn các dữ liệu nhiều chiều một cách hiệu quả. Kỹ thuật phổ biến nhất là biểu đồ phối hợp song song. Với một biểu đồ loài này, các trục được đặt song song với nhau cho mỗi biến và mỗi thể hiện dữ liệu một được thẳng được vẽ nối các giá trị trên



Hình 6.10: Biểu diễn dữ liệu 3 chiều với ma trận biểu đồ phân bố hai chiều.

mỗi trục. Hình 6.11 là biểu diễn bằng biểu đồ phối hợp song song cho dữ liệu ở bảng 6.1. Khi các trục không được đặt song song với nhau mà theo vòng tròn thì biểu đồ được gọi là biểu đồ ra đa hoặc biểu đồ sao. Hình 6.12 biểu diễn dữ liệu ở bảng 6.1 với biểu đồ sao.

Hình 6.11: Biểu diễn dữ liệu đa chiều với biểu đồ phối hợp song song.



Hình 6.12: Biểu diễn dữ liệu đa chiều với biểu đồ ra đa (biểu đồ sao).

6.5. CÁC BIỂU ĐỒ ĐỘNG

Các biểu đồ động là các biểu đồ thay đổi diện mạo khi tương tác với người dùng. Sử dụng biểu đồ động là cách hiệu quả để biểu diễn trực quan các xu hướng và tiết kiệm không gian thay vì sử dụng một loại các biểu đồ tĩnh. Theo định nghĩa, các biểu đồ động chỉ có thể được cài đặt trong các hệ thống thông tin được máy tính hóa và không thể sử dụng trong các báo cáo quản lý truyền thống dựa trên bản giấy.

Kỹ thuật đầu tiên là kỹ thuật cọ vẽ (brushing) được mô tả bởi Becker và Cleveland năm 1987 sử dụng chuột máy tính để tương tác với biểu đồ động. Khi dùng chuột trỏ đến một chấm nào đó hoặc một khu vực trong biểu đồ thì biểu đồ sẽ thay đổi hình dạng. Ví dụ, nếu tương tác với biểu đồ trong hình 6.8, khi trỏ đến một chấm nào đó, chấm đó sẽ được làm nổi bật so với các chấm khác. Hoặc khi di chuyển cọ vẽ trên biến chuỗi thời gian, biểu đồ sẽ hiển thị thông tin chi tiết về thời gian tại vị trí con trỏ.

Kỹ thuật thứ hai là chú thích tương tác. Nếu có qua nhiều thể hiện hoặc giá trị phân loại, chú thích có thê được sử dụng trong tương tác. Mỗi giá trị phân loại có thể được hiển thị hoặc ẩn. Ví dụ, trong biểu đồ phối hợp song song, chúng ta có thể hiển thị hoặc ẩn các đường của các thể hiển cụ thể để tập trung vào các thể hiện chúng ta quan tâm.

Kỹ thuật chú thích tương tác là một ví dụ cụ thể của kỹ thuật tổng quát gọi là kỹ thuật phủ (overlay). Kỹ thuật phủ tổ chức các bảng và các biểu đồ thành các lớp và cho phép và cho phép hiển thị hoặc ẩn các lớp. Một lớp có thể được xếp chồng lên lớp khác để cung cấp thêm thông tin hoặc có thể không được hiển thị để bớt thông tin được trình bày.

6.6. MÀU SẮC VÀ CÁC HIỆU ỨNG HÌNH ẢNH KHÁC

Sử dụng mầu sắc trong hiển thị có thể đạt được các hiệu ứng mạnh mẽ. Thứ nhất là hiệu ứng thông tin. Mầu sắc có thể được sử dụng để tăng thêm thông tin cho một biểu đồ. Ví dụ, mầu sắc có thể được sử dụng để chỉ thị các giá trị ‘tốt’ hoặc ‘xấu’ trên các biểu đồ bằng các giá trị mầu tương ứng xanh và đỏ. Chúng ta cũng đã tìm hiểu về biểu đồ bong bóng được sử dụng kết hợp với khuynh độ mầu ở đó các giá trị mầu có thể được sử dụng để biểu diễn giá trị dữ liệu ở một chiều thứ tư.

Hiệu ứng thứ hai là hiệu ứng cô lập. Một mầu có thể được sử dụng để cô lập một giá trị dữ liệu từ các giá trị khác để thu hút sự chú ý đến giá trị dữ liệu đó. Hiệu ứng này chỉ hiệu quả nếu sử dụng các mầu một cách thưa thớt. Nếu sử dụng nhiều mầu thì các mầu sẽ cạnh tranh nhau về sự chú ý và hiệu ứng cô lập không còn nữa.

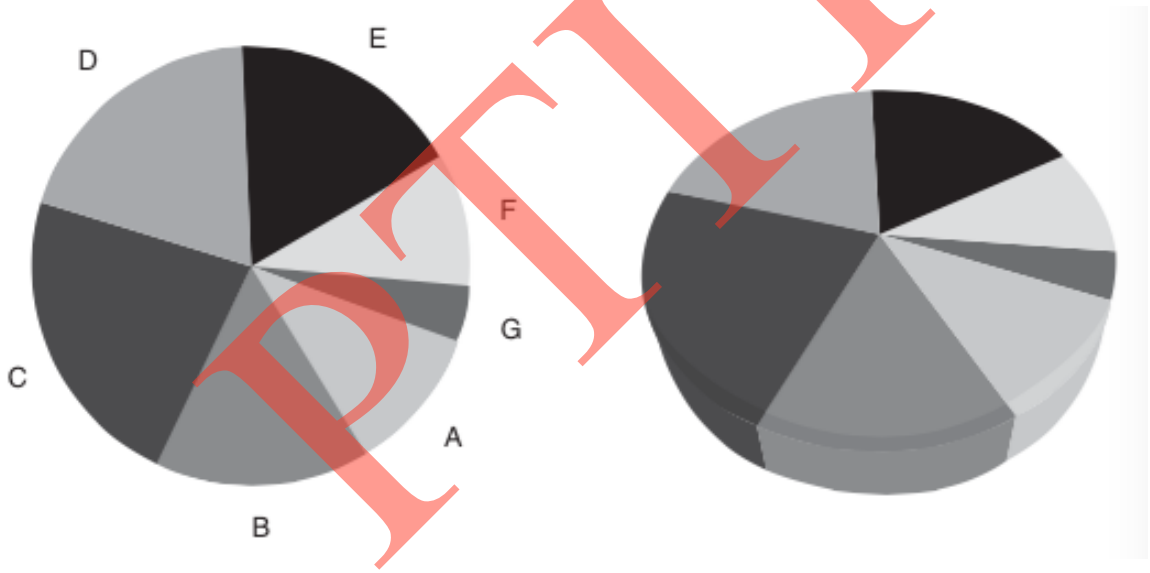
Hiệu ứng thứ ba là hiệu ứng thẩm mỹ. Hiệu ứng thẩm mỹ có được khi thêm mầu sắc gọi là sự hài hòa về mầu sắc. Sự hài hòa về mầu sắc là kết quả của sự kết hợp các mầu cho cảm giác trình tự và cân bằng thị giác.

Việc chọn một mầu có thể sinh ra cả ba hiệu ứng nêu trên đồng thời do đó việc sử dụng mầu sắc cần phải được thực hiện một cách cẩn trọng. Một cách để chọn mầu là sử dụng ‘bánh xe mầu’. Một bánh xe mầu hiển thị một số các mầu lân cận nhau tạo thành một vòng tròn. Để đạt được các hiệu ứng hài hòa, chúng ta cáo thể chọn các mầu tương tự nhau có nghĩa là ở cạnh nhau

phía bên phải trong bánh xe mầu. Để đạt được các hiệu ứng tương phản, chúng ta chọn các mầu bù nhau là các mầu nằm đối diện nhau trong bánh xe mầu.

Bất kể những mầu nào được chọn, không nên sử dụng quá nhiều mầu và sử dụng mầu một cách thận trọng. Thường không nên sử dụng nhiều hơn ba hoặc bốn mầu. Lý do là nếu sử dụng nhiều mầu thì ba hiệu ứng nêu ở phần trên sẽ nhanh chóng cạnh tranh nhau và hiệu ứng định truyền tải đến người dùng sẽ bị mất đi. Tương tự, hiệu ứng thẩm mỹ sẽ bị mất đi do sự mất hài hòa khi sử dụng quá nhiều mầu.

Các hệ thống thông tin quản lý thương mại hóa thường cho phép người dung trang trí các biểu đồ với nhiều mầu sắc và các hiệu ứng đẹp mắt. Tuy nhiên việc sử dụng không hợp lý các hiệu ứng này có thể dẫn đến các kết quả không mong muốn. Lý do là các hiệu ứng có thể làm méo sự trình bày chính xác dữ liệu, đặc biệt là khi biểu diễn tỷ lệ. Hình 6.13 mình họa hiệu ứng méo trong trình bày chính xác dữ liệu khi sử dụng biến thể biểu đồ tròn 3D. Chúng ta thấy biểu đồ 3D nhìn bắt mắt hơn nhưng biểu diễn trực quan các giá trị cho B và E bị bóp méo trong phiên bản 3D. Mặc dù giá trị B và E là bằng nhau nhưng trong phiên bản biểu đồ 3D ấn tượng thị giác là B lớn hơn E.



Hình 6.13. Sự bóp méo trực quan thị giác về biểu diễn dữ liệu khi sử dụng biểu đồ dạng tròn (pie) 3D.

Một lý do khác là không nên trang trí các biểu đò là bản thân biểu đồ sẽ thu hút sự chú ý nhiều hơn so với dữ liệu được cung cấp là cơ sở cho xây dựng biểu đồ. Tóm lại, khi càng thêm nhiều hiệu ứng hình ảnh, dữ liệu gốc sẽ càng bị đánh mất sự chú ý.

CHƯƠNG 7: HỖ TRỢ RA QUYẾT ĐỊNH QUẢN LÝ

- *Xác định các chỉ số đánh giá hiệu năng cho các thực thể quản lý*
- *Tổ chức và nhóm các KPI trong các khung khác nhau*
- *Phác thảo các kỹ thuật để hiển thị các biến động của KPI*
- *Hiểu chiến lược và các kỹ thuật áp dụng cho quá trình ra quyết định*

7.1. Ý NGHĨA CỦA VIỆC RA QUYẾT ĐỊNH QUẢN LÝ

Người quản lý dành nhiều thời gian nghiên cứu hiệu quả của các thực thể quản lý và các mối quan hệ giữa chúng để đưa ra các hành động phù hợp trong trường hợp hiệu quả hoạt động giảm sút xuống dưới một ngưỡng nào đó. Một hệ thống thông tin thường không chỉ hiển thị hiệu quả hoạt động mà còn biểu diễn các thay đổi về hiệu quả hoạt động của tổ chức theo thời gian. Hệ thống thông tin quản lý cần phải đưa ra cảnh báo khi hiệu quả hoạt động thay đổi một cách không mong đợi. Trên cơ sở các cảnh báo này, người quản lý quyết định xem hành động khắc phục có cần thiết hay không.

Hiệu quả của hoạt động của một tổ chức được đại diện bởi tập các chỉ số hiệu quả hoạt động chính. Việc theo dõi một tập lựa chọn các chỉ số hiệu quả hoạt động chính là một loại quyết định quản lý cần được xem xét.

7.2. XÁC ĐỊNH CÁC CHỈ SỐ ĐÁNH GIÁ HIỆU QUẢ HOẠT ĐỘNG CHÍNH KPI

Một trong những tính năng được mong đợi từ một hệ thống thông tin quản lý là khả năng hiển thị các chỉ số hiệu quả hoạt động chính trong một báo cáo tóm tắt mức cao được trình bày một cách xúc tích. Báo cáo quản lý đó gọi là các thẻ điểm (scorecard) ở đó các điểm của tổ chức sẽ được nhập.

Một khái niệm khác ngày càng được sử dụng phổ biến đó là bảng biểu đồ thông tin (dashboard) hoặc đơn gian là bảng biểu đồ. Khái niệm này được sử dụng để hàm ý một thẻ điểm tương tác, một thẻ điểm với chức năng tương tác để thay đổi hiển thị KPI trên thẻ điểm.

Việc xác định các chỉ số hiệu quả hoạt động chính là một phần quan trọng trong thiết kế hệ thống thông tin quản lý đòi hỏi việc suy xét cẩn trọng về cách thức các thực thể quản lý được quản lý.

Một cách để bắt đầu xác định các chỉ số hiệu quả hoạt động chính là các phép đo có ý nghĩa nhất định đối với ‘sức khỏe’ của các thực thể và các mối quan hệ đang được quản lý. Ví dụ, khi quản lý một tổ chức bán hàng, một chỉ số phản ánh tình trạng sức khỏ cả tổ chức bán hàng là tổng lợi tức bán trong một tháng nào đó.

Cách thứ hai để xác dịnh các chỉ số hiệu quả hoạt động chính là xem xét vòng đời thực thể quản lý và các giai đoạn khác nhau của thực thể. Sau đó chúng ta có thể nghiên cứu các điều kiện chuyển dịch giữa các giai đoạn của vòng đời và xác định các chỉ số cho biết bao nhiêu thể

hiện của một thực thể dịch chuyển từ một giai đoạn này sang giai đoạn khác. Các tốc độ chuyển đổi (ví dụ, chuyển đổi từ khách hàng triển vọng sang khách hàng thực sự) là cac ví dụ về các chỉ số hiệu quả hoạt động có thể có được theo cách tiếp cận này.

Việc nhận diện các chỉ sô hiệu quả hoạt động chính có thể lấy cảm hứng từ chiến lược dài hạn được xác định bởi nhóm quản lý. Nhiều tổ chức liên kết các chỉ số hiệu quả hoạt động chính với các định hướng chiến lược một cách trực tiếp. Ví dụ, định hướng chiến lược là tăng trưởng thị trường cho một số sản phẩm nào đó, thì chỉ số hiệu quả hoạt động chính sẽ là thị phần của tổng lợi tức bán so với tổng bán của thị trường. Nếu chiến lược dài hạn là tăng trưởng lợi tức của các thị phần các sản phẩm phi thực phẩm so với sản phẩm thực phẩm, thì chỉ số hiệu quả hoạt đồng chính sẽ là tỷ lệ các sản phẩm phi thực phẩm tạo ra tổng lợi tức bán hàng.

Xác định mối liên hệ với chiến lược dài hạn thường khá khó khăn. Có rất nhiều các mẫu được sử dụng. Trong số đó, Thẻ điểm cân bằng (BSC- Balanced Scorecard) là mẫu được sử dụng phổ biến cho các tổ chức muốn tối đa hóa lợi nhuận. Với các tổ chức phi lợi nhuận, cần phải thay đổi mẫu thẻ điểm.

Cái được cân bằng trong thẻ điểm cân bằng là nó không chỉ tập trung vào các chỉ số hiệu quả hoạt động tài chính như là lợi tức bán hàng và lợi nhuận. Ngoài các chỉ số tài chính còn các nhóm chỉ số triển vọng khác.

- *Triển vọng tài chinh*: Đây là triển vọng tài chính dưới quan điểm kế toán. Nó bao gồm các dữ liệu tổng hợp tài chính thường thấy trong các bảng cân đối và các số liệu về lợi nhuận và lỗ.

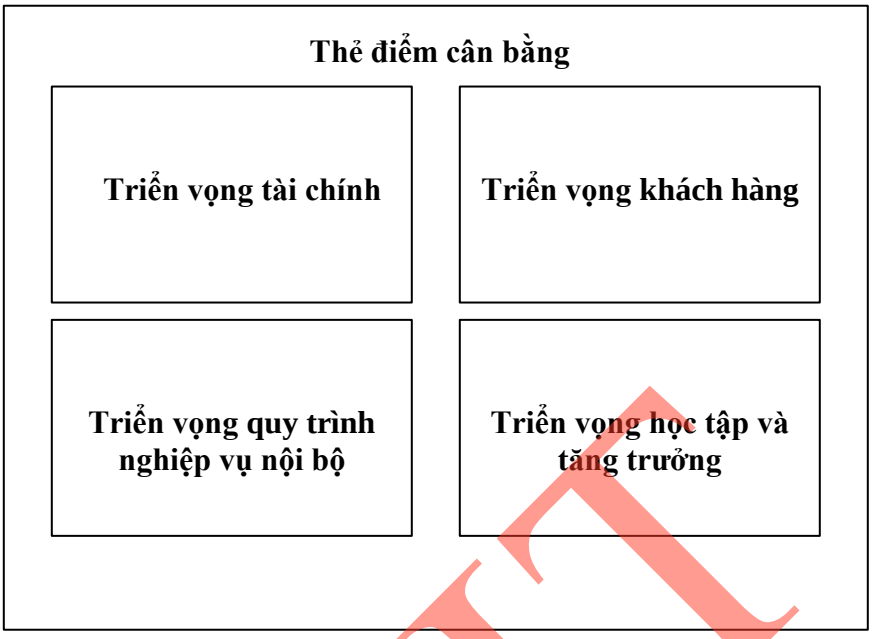
  Các tỷ lệ có thể nhóm trong triển vọng tài chính bao gồm lợi nhuận trên đầu tư.

- *Triển vọng khách hàng*: Đây là triển vọng dưới quan điểm tiếp thị. Các chỉ số hiệu quả hoạt động quan trọng đối với khách hàng sẽ được xem xét. Ví dụ, thời gian bàn giao sản phẩm trung bình. Triển vọng khách hàng cũng xem xét các chỉ số liên quan đến khách hàng như sự hài lòng của khách hàng.

  Các tỷ số có thể nhóm trong triển vọng khách hàng bao gồm các tỉ lệ chuyển đổi. Ví dụ, phần trăm khách hàng triển vọng trở thành khách hàng thực sự.

- *Triển vọng quy trình nghiệp vụ nội bộ*: Đây là triển vọng liên quan đến quản lý điều hành. Các chỉ số hiệu quả hoạt động chính có thể bảo gồm các mức tồn kho kiểm kê, thời gian cần thiết để lắp ráp một sản phẩm v.v.
- *Triển vọng học hỏi và tăng trưởng*: Đây là triển vọng liên quan đế quản lý nguồn nhân lực. Nó có thể bao gồm các tỷ lệ như là doanh thu trên nhân viên v.v và các chỉ số liên quan đến phát triển đội ngũ.

BSC thường được trình bày như một báo cáo tóm tắt một trạng với các triển vọng được sắp xếp đối xứng để đảm bảo hiệu ứng thẩm mỹ. Hình 7.1 là một ví dụ thẻ điểm cân bằng BSC.



Hình 7.1. Thẻ điểm cân bằng.

Ngoài khung thẻ điêm cân bằng được sử dụng để nhận dạng các chỉ số hiệu quả hoạt động chính. Một số tác giả khác như Peter Drucker cũng đề xuất các phân loại các chỉ số hiệu quả hoạt động chính như sau:

- *Thông tin nền tảng*: Đây là thông tin chẩn đoán của các thực thể và các mối quan hệ quản lý đang được quản lý. Ví dụ, lợi tức mà các nhóm bán hàng tạo ra.
- *Thông tin năng suất*: Thông tin chẩn đoán này cho chúng ta biết về năng suất của các thực thể và các mối quan hệ quản lý. Ví dụ, lợi tức trung bình tạo ra bởi một nhân viên bán hàng.
- *Thông tin năng lực chuyên môn*: Đây là thông tin về hiệu quả của năng lực chuyên môn tốt nhất của tổ chức. Theo một cách nào đó, đây là khía cạnh chiến lượng của thông tìn nền tảng ở đó tập trung vào các định hướng chiến lượng dài hạn của công ty.
- *Thông tin phân bổ nguồn lực*: Đây là thông tin chẩn đoán về các nguồn lực được phân bổ và hiệu quả của nguồn lực được phân bổ.

Khi đã nhận diện được các chỉ số hiệu quả hoạt động chính, cần phân biệt giữa các chỉ số về quá khứ (lagging) và các chỉ số về tương lai (leading). Chỉ số về quá khứ là các chỉ số của các sự kiện xảy ra trong quá khứ. Ví dụ, lượng bán hàng là một chỉ số về quá khứ ví nó cho biết sự thành công của các giao dịch bán hàng trong quá khứ. Chỉ số về tương lai các chỉ số của các sự kiện sẽ xảy ra trong tương lai. Ví dụ, số các khách hàng triển vọng là một chỉ số về tương lai. Có các chỉ số vừa có thể là chỉ số về tương lai vừa có thể là chỉ số về quá khứ.

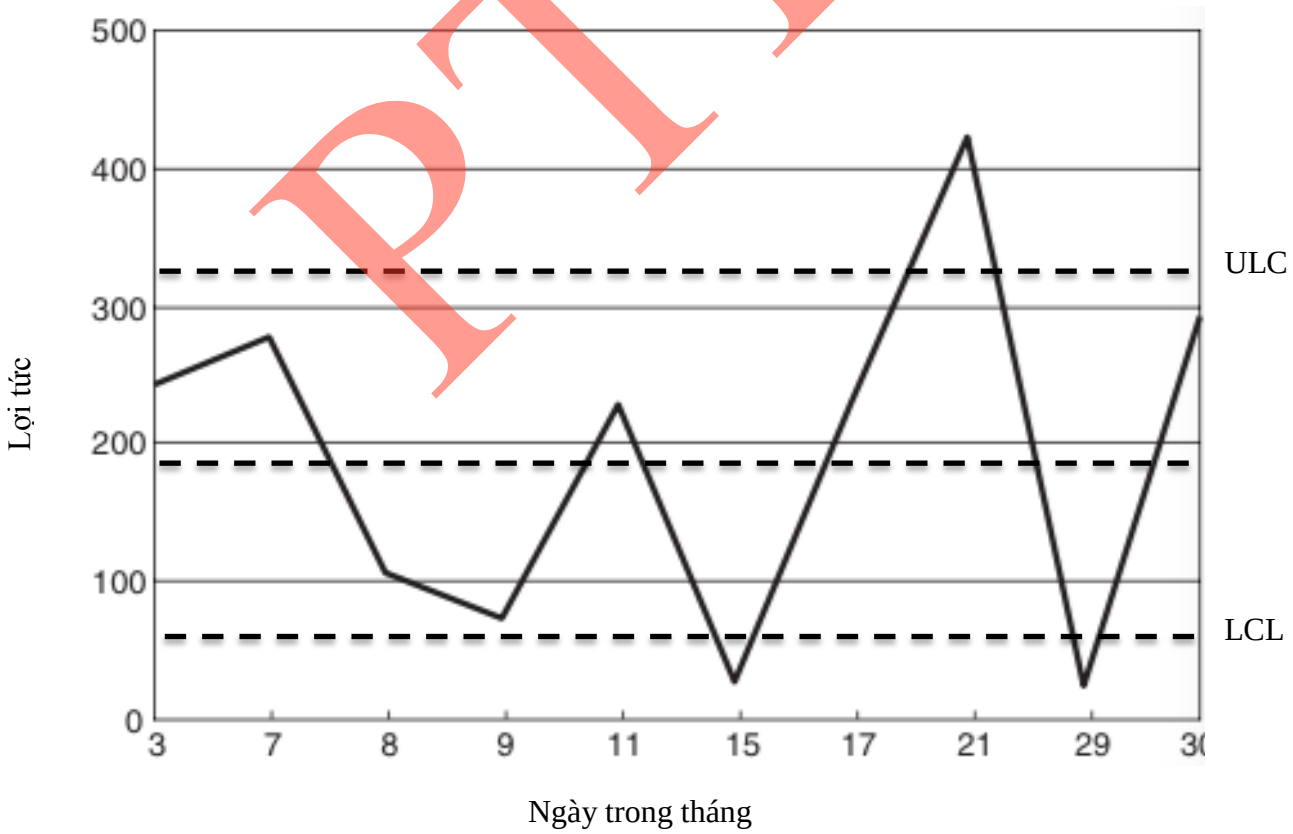
Các chỉ số hiệu quả hoạt động có thể được tổ chức thành các kiến trúc phân cấp. Các chỉ số ở mức thấp (ví dụ, lượng bán hàng từ nhóm Alpha) có thể được đưa vào các chỉ số ở mức cao hơn (ví dụ, lượng bán hàng của tất cả các nhóm). Các hệ thống trí tuệ doanh nghiệp tiên tiến cho phép định nghĩa các mục tiêu chiến lược mức cao (như là 'tăng tỷ lệ chuyển đổi khách hàng triển vọng thành khách hàng thực sự') và nhóm các chỉ số liên quan cần có để thấy mục tiêu đang được đáp ứng ở mức thấp hơn của kiến trúc phân cấp. Do đó, có thể đào sâu từ mục tiêu chiến lược thành các chỉ số mức thấp.

7.3. CÁC KỸ THUẬT GIÁM SÁT KPI

7.3.1. Thêm băng thông (bandwidth)

Một chỉ số hiệu quả hoạt động chính thường được khái niệm hóa như một giá trị dữ liệu tổng hợp, thường là một giá trị dẫn suất. Chúng ta có thể biểu diễn trực quan các KPIs theo các nguyên lý và phương pháp thảo luận ở chương 6.

KPIs thường được theo dõi theo thời gian do thường bản thân một KPI không có nhiều ý nghĩa và là biến động của KPI có ý nghĩa quan trọng. Các nghiên cứu về biến động KPIs theo thời gian gọi là phân tích xu hướng. Để nghiên cứu các xu hướng cần xem xét các thay đổi của KPI theo thời gian. Biểu đồ sử dụng thường có dạng là biểu đồ chuỗi thời gian. Để hỗ trợ diễn dịch biểu đồ như thế, băng thông có thể được thêm vào các biểu đồ. Hình 7.2 là mộ ví dụ về thêm băng thông vào biểu đồ.



Hình 7.2:Giới hạn kiểm soát dưới (LCL) và giới hạn kiểm soát trên (UCL).

Hình 7.2 cho thấy biểu đồ chuỗi thời gian chuẩn. Đường đứt nét ở giữa là giá trị trung bình cho 12 tháng. Các đường đứt nét ở phía trên và phía dưới tương ứng là các đường giới hạn kiểm soát trên và các đường giới hạn kiểm soát dưới. Một biểu đồ chuỗi thời gian có một băng như trên được gọi là biểu đồ quá trình. Các đường giới hạn cho biết các giá trị KPI nào là bình thường (nằm trong băng thông) và các giá trị nào là bất thường (nằm ngoài băng thông). Để tính các giá trị cho các đường kiểm soát có thể sử dụng giá trị trung bình và độ lệch chuẩn của KPI trong một khoảng thời gian nhất định. Một giới hạn thường được thiết lập bằng giá trị trung bình cộng/trừ ba lần độ lệch chuẩn. Bất kỳ giá trị nào ở bên ngoài băng thông một cách thống kê được coi là các ngoại lai.

Hình 7.2 biểu diễn băng thông ổn định do nó không hiệp biến với giá trị của KPI. Có thể định nghĩa một băng thay đổi khi KPI thay đổi gọi là băng thông động. Ví dụ, thay vì định nghĩa các giới hạn sử dung giá trị trung bình ổn định trong một khoảng thời gian cố định, chúng ta có thể tính toán loại các giới hạn dựa trên các giá trị trung biinhf biến đổi được cập nhật với các giá trị KPI mới nhất. Các giá trị trụng bình thay đổi như thế thường được sử dụng trong dữ liệu trao đổi cổ phiếu để theo dõi biến động giá cổ phiếu.

Ý tưởng đăng sau việc thêm băng thông vào là để luôn có biến động thống kê trong các chỉ số hiệu quả hoạt động chính nhưng đó không phải là trọng yếu cho đến khi nó di chuyển ra ngoài biên thiết lập bởi các giới hạn kiểm soát trên và giới hạn kiểm soát dưới. Nhà quản lý có thể sau đó quyết định có hành động khắc phục khi KPI tiếp cận hoặc vượt quá các giới hạn. Lợi ích quan trọng của việc thêm băng thông là giúp dễ dàng phát hiện các hiệu quả cực trị.

7.3.2. Thêm chỉ số so sánh

Để đánh giá biên độ của một giá trị nào đó của một chỉ số hiệu quả, giá trị đó thường được so sánh với giá trị liên quan. Một hệ thống thông tin quản lý cần thực hiện các so sánh với một dải giá trị liên quan. Phân tích so sánh như thế được gọi là phân tích hiệp biến hoặc phân tích khác biệt.

Có ít nhất ba loại phân tích so sánh các thể được xác định cho bất kỳ một KPI. Các loại phân tích đó là:

- *So sánh lịch sử*: Giá trị của mội KPI được so sánh với các giá trị của KPI đó trong lịch sử.
- *So sánh mục tiêu*: Giá trị của một KPI được so sánh với giá trị mục tiêu mong muốn. Giá trị mục tiêu có thể được thiết lập trước hoặc có thể được tính toán động.
- *So sánh cạnh tranh*: Giá trị của một KPI được so sánh với giá trị của KPI tương tự của một hay nhiều tổ chức cạnh tranh.

Với phân tích so sanh, các giá trị mô tả thương không được quan tâm nhiều so với mức độ thay đổi giá trị so với giá trị trong quá khứ. Ví dụ , báo cáo về mức cổ phiếu của một công ty tăng lên 20% có ý nghĩa hơn là mức cổ phiếu năm ngoái là 64 năm nay là 76,8. Nếu sự khác biệt là yếu tố quan tâm thì cần phải tránh quá tải thông tin bằng cách báo cáo thay đổi chứ không phải là hai giá trị được sử dụng để tính sự khác biệt.

Biểu diễn trực quan các so sánh có thể được thực hiện hiệu quả bằng các biến thể biểu đồ thanh side-by-side. Một cách khác là hiển thị các giá trị so sanh với KPI trong biểu đồ quá trình. Các biến động hình thành trên các giá trị so sánh có thể được quan sát và sự khác nau so với biến động của KPIs.

7.3.3. Ngoại lệ

Một kỹ thuật quan trọng trong báo cáo các KPIs được gọi là *ngoại lệ*. Ý tưởng của kỹ thuật là chỉ hiển thị các dữ liệu cực trị. Với kỹ thuật này, các dữ liệu khác bị loại bỏ và chỉ còn lại các ngoại lệ nhằm loại bỏ các thông tin không quan tâm. Khi sử dụng kỹ thuật ngoại lệ, cần phải thiết kế các tiêu chí cho các giá trị dữ liệu được coi là ngoại lệ. Ví dụ, các ngoại lai thống kê có thể được coi là các ngoại lệ. Ngoài ra, có thể xây dựng các tiêu chí khác cho các ngoại lệ tùy thuộc vào ưng dụng. Ví dụ, đơn vị được đánh giá tốt nhất các tháng hoặc tháng ở đó các đơn vị làm việc kém nhất v.v.

7.3.4. Phân tích độ nhạy

Một kỹ thuật khác được sử dụng để theo dõi KPI là kỹ thuật phân tích độ nhạy. Phân tích độ nhạy mô phỏng các thay đổi xảy ra trong một KPI nếu dữ liệu cơ sở nào đó thay đổi.

Chúng ta đã biết một KPI thường được dẫn suất từ dữ liệu cơ sở. Do đó, để tính toán KPI cần có một số các giá trị đầu vào. Nếu các giá trị đầu vào thay đổi thì giá trị KPI cũng thay đổi theo. Ví dụ, lợi tức bán hàng tổng phụ thuộc vào một số các đơn hàng mà tổ chức bán hàng cố đặt được. Một phân tích độ nhạy sẽ hiển thị hiệu ứng của các thay đổi này đến giá trị của KPI. Loại phân tích độ nhạy này được biết đến như là phân tích 'what-if'.

Cách nghĩ thông thường về kỹ thuật này là thay đổi đầu vào và xem xét ảnh hưởng đến đầu ra. Tuy nhiên, phân tích này cũng có thể được thực hiện theo chiều ngược lại ở đó đầu ra được cố định và tìm các giá trị đầu vào để có được đầu ra cụ thể. Loại phân tích ở đó đầu ra mục tiêu được thay đổi và xem xét cách đầu vào thay đổi được gọi là phân tích tìm kiếm mục tiêu (goal-seeking).

7.4. MA TRẬN QUYẾT ĐỊNH

Giả sử ta muốn chọn một nhà sản xuất một vật liệu cụ thể, ví dụ gỗ, để sử dụng trong quy trình sản xuất. Bảng 7.1 là các lựa chọn khác nhau. Làm thế nào để ra quyết định chọn nhà cung cấp?

Có thể thấy biểu diễn trong bảng 7.1 giốn với các bảng giải chuẩn. Việc xây dựng ma trận quyết định theo cùng nguyên lý cơ bản của việc cấu trúc hóa dữ liệu ở chương 2. Ẩn đằng sau bảng 7.1 là các thực thể Nhà cung cấp, Vật liệu, Vùng và Đánh giá chất lượng.

Một hệ thống thông tin cần có khả năng hiểu thị ma trận quyết định như ở bảng 7.1 theo một định dạng dễ đọc và cho phép người dùng thao tác với các hàng (các lựa chọn) và các cột (các thuộc tính).

Điểm chú ý tiếp theo về Bảng 7.1 là chúng ta cũng có thể tập trung vào các khả năng sắp xếp các lựa chọn theo thuộc tính và khả năng xếp hạng bị giới hạn bởi thang đo của thuộc tính. Ví dụ, thuộc tính 'Vùng' ở thang đo danh định và không thể sử dụng để xếp hạng. Chúng ta

không thể nói vùng A 'cao hơn' hay 'thấp hơn' vung B. Thuộc tính 'Chất lượng' ở thang đo thứ tự. Các thuộc tính 'Khoảng cách đến nhà máy', 'Khoảng cách đến kho' và 'Giá' ở thang đo tỷ lệ.

Bảng 7.1. Tìm kiếm và lựa chọn nhà cung cấp tốt nhất.

#	Vùng	Chất lượng	K. Cách N. Máy	K. Cách Kho	Giá
1	A	2 star	60	100	120
2	B	3 star	40	130	100
3	B	4 star	100	150	200
4	A	2 star	400	110	70
5	B	3 star	50	100	150
6	A	4 star	50	140	100
7	B	2 star	200	50	40
8	A	4 star	100	110	40
9	A	2 star	300	100	30
10	B	3 star	400	110	70
11	A	2 star	200	75	80
12	A	2 star	300	110	50

7.5. CÁC CHIẾN LƯỢC RA QUYẾT ĐỊNH

Quyết định về một lựa chón tốt nhất trong ma trận quyết định như ở Bảng 7.1 liên quan đến thực hiện nhiều phép so sánh. Chúng ta bắt đầu bằng việc tập trung vào một thuộc tính cụ thể trong ma trận quyết định. Ví dụ, giá của gỗ từ nhà cung cấp số 2 là 100. Để hiểu giá trị này là tốt hay xấu ta phải thực hiện hai loại so sánh. Loại so sánh đầu tiên tập trung vào thuộc tính, ta phải xem giá của các nhà cung cấp khác xem đắt hơn hay rẻ hơn. Đó là phép so sánh theo chiều đứng dựa trên cột. Loại so sánh khác tập trung vào các lựa chọn, tả phải xem xét các giá trị khác của nhà cung cấp cụ thể và cân nhắc yếu tố bù lại giá. Đó là phân tích theo chiều ngang dựa trên hàng.

Có nhiều cách tiếp cận giải quyết vấn đề quyết định đa lựa chọn, đa biến. Các cách tiếp cận này được xem là các chiến lược ra quyết định. Các chiến lược có thể liên quan đến các so sánh cặp đội. Nghĩa là xem xét các lựa chọn theo cặp, so sánh chúng và loại bỏ một lựa chọn không được ưa thích. Tiến trình so sánh tiếp tục cho đến khi có được lựa chọn cuối cùng. Một số chiến lược ra quyết định được mô tả sau đây:

- *Chiến lượng ra quyết định gia tăng trọng số (Weighted Additive Decision Strategy - WADD)*: Ý tưởng đằng sau chiến lược này là gán các trọng số cho mỗi thuộc tính và sử dụng các trọng số để tính toán một điểm tổng cho từng lựa chọn trên cơ sở đó chọn lựa chọn có điểm tốt nhất.

  Ví dụ, giả sử chúng ta muốn xem xét hai thuộc tính: 'Khoảng cách đến kho hàng' và 'Khoảng cách đến nhà máy'. Ta gán khoảng cách đến nhà máy một trọng số là 70% và khoảng cách đến kho hàng là 30%. Điểm cho nhà cung cấp số 1 là 0.70x60+0.30x100 = 72. Tương tự, điểm cho nhà cung cấp số 2 là 67. Nhà cung cấp số 2 trong trường hợp này

tốt hơn nhà cung cấp số 1 vì trong trường hợp này điểm tốt hơn là điểm thấp nhất (khoảng cách càng ngắn đến kho hàng và nhà máy càng tốt). Chiến lược này không thể áp dụng cho các thang khoảng, thứ tự hoặc danh định. Lý do là ta phải có thể cộng và nhân các giá trị thuộc tinh do đó các thuộc tính phải ở thang tỷ lệ. Để đưa các thuộc tính ở thang khác bào xem xét, cần phải ánh xạ từng giá trị thuộc tính vào một giá trị tiện ích sử dụng hàm tiện ích. Sau đó, có thể sử dụng các giá trị hiệu dụng trong chiến lược loại này.

- *Chiến lược ra quyết đinh trọng số tương đương (Equal weights decision strategy – EQW)*: Đây là phiên bản đơn giản hóa của chiến lược WADD ở đó mỗi thuộc tính được giả thiết có trọng số bằng nhau. Trong trường hợp này, không cần thiết phải gán các trọng số cho các thuộc tính. Với mỗi lựa chọn, ta sẽ tính tổng tiện ích của giá trị và lựa chọn với điểm tốt nhất sẽ được chọn.

  Ví dụ, xem xét khoảng cách đến nhà máy và khoảng cách đến kho. Nhà cung cấp số 1 sẽ có điểm là 60+100 = 160. Nhà cung cấp số 2 có điểm là 40+130=170. Do đó, nhà cung cấp số một được lựa chọn sử dụng chiến lược EQW.

- *Chiến lược khác biệt gia tăng (Additive Difference Strategy – ADIFF)*: Ý tưởng cơ bản là cộng tổng các khác biệt và sinh ra điểm khác biệt. Ví dụ khác biệt về khoảng cách tới nhà máy giữa nhà sản xuất số 1 và nhà sản xuất số 2 là -20. Khác biệt về khoảng cách đến kho chứa giữa nhà cung cấp số 1 và nhà cung cấp số 2 là +30.
  Điểm khác biệt là 0.70x(-20)+ 0.30x30=(-5). Có nghĩa là nhà cung cấp số 2 là tốt hơn -5 so với nhà cung câp số 1.

- *Chiến lược đa số của chiều xác nhận (Majority of confirming dimensions-MCD)*: Đây là biến thể của chiến lược ADIFF được đơn giản hóa để áp dụng với các thang thứ tự và phi thông ước. Bắt đầu với cặp đầu tiên, đếm số thuộc tính mà một lựa chọn tốt hơn lựa chọn kia. Ta cũng có thể đếm số các thuộc tinh mà một lựa chọn tồi hơn lựa chọn kia. Nếu số các lựa chọn tốt hơn là đa số thì tiến hành với lựa chọn tiếp theo.

- *Chiến lược ra quyết đinh hy sinh (Sacrificing Decision Strategy-SAT)*: Khái niệm hy sinh được đề xuất bởi Herbert Simon để miêu tả các loại quyết định ở đó chúng ta không nhắm tới việc chọn một lựa chọn tốt nhất mà là một lựa chọn đủ tốt. Khi một lựa chọn đủ tốt, chúng ta sẽ dừng tìm kiếm lựa chọn tốt hơn. Chiến lược này được gọi là hy sinh do sự tối ưu bị hy sinh. Các mức ngưỡng cho mốt thuộc tính phải được định nghĩa trước trong chiến lực này.

  Ví dụ, chúng ta định nghĩa các mức ngưỡng cho các nhà cung cấp trong Bảng 7.1. Giả sử khoảng cách đến nhà máy là 120 sẽ đủ tốt với giá thấp hơn 150. Chúng ta sẽ hy sinh vật liệu 3 sao. Vùng không quan trọng. Trong trường hợp đó nhà cung cấp đáp ứng các giá trị ngường là nhà cung cấp số 5.

  Với chiến lược hy sinh, rủi ro là bỏ lỡ các lựa chọn tốt hơn. Trong ví dụ trên, nhà cung cấp số 6 cũng đáp ứng các giá trị ngưỡng và tốt hơn nhà cung cấp số 5. Do đó, thứ tự đóng vai trò quan trọng trong chiến lược này. Với chiến lược này, các lựa chọn được so sánh với các giá trị ngưỡng thay vì so sánh với nhau nên không cần phải đánh giá hết 12 lựa chọn trước khi quyết định.

- *Tần số của các đặc trưng tốt và xấu*: Chiến lược này là mở rộng của chiến lược SAT. Trước hết, phải định nghĩa các giá trị ngưỡng cho các thuộc tính. Thay vì quyết định nếu các thuộc tính đáp ứng được các giá trị ngưỡng, số các thuộc tính dương sẽ được đếm và chuyển qua từng lựa chọn nếu số đó được tăng.
- *Tự từ điển (Lexicographic-LEX)*: Các chiến lược đã trình bày đều dựa trên các lựa chọn. Có nghĩa là ta có xu hương xem xét các lựa chọn, và các thuộc tính quan tâm sau đó quyết định giá trị của lựa chọn. Ở khía cạnh ma trận ra quyết định, ta thường có khuynh hướng xem xét các hàng. Hai chiến lược còn lại dựa trên thuộc tính. Có nghĩa là các thuộc tính sẽ được xem xét trước.

  Chiến lược dựa trên thuộc tính đầu tiên là chiến lược tự từ điển. Trước hết cần quyết định thuộc tính quan trọng nhất. Sau đó chọn lựa chọn tốt nhất dựa trên thuộc tính này. Nếu có hai hoặc nhiều hơn lựa chọn cạnh tranh nhau, tiến hành với thuộc tính quan trọng thứ hai. Từ tập con các thuộc tính quan trọng, lựa chọn tốt nhất sẽ được chọn dựa trên thuộc tính quan trọng tiếp theo. Quá trình này lặp lại cho đến khi chọn được lựa chọn tốt nhất.

  Ví dụ với ma trân ra quyết định trong Bảng 7.1, giả sử thuộc tính quan trọng nhất là là chất lượng. Trong trường hợp đó, tất cả các nhà cung cấp với chất lượng tốt nhất sẽ được lựa chọn đó là các nhà cung cấp số 3, số 6 và số 8. Cần phải chọn thuộc tính quan trọng thứ hai và đó có thể là khoảng cách đến kho hàng. Kết quả lựa chọn tốt nhất là nhà cung cấp số 8.
- *Loại trừ bằng khía cạnh (Elimination by Aspect-EBA)*: Khái niệm khía cạnh được sử dụng đồng nghĩa với thuộc tính. Chiến lược này kết hợp các phần tử của chiến lược hy sinh với chiến lược tự từ điển. Giống như chiến lược tự từ điển, chiến lược này xem xét các thuộc tính trước tiên và xếp hạng các thuộc tính này theo thứ tự quan trọng. Giống như chiến lược hy sinh, chiến lược này loại trừ các lựa chọn sử dụng các gí trị ngưỡng. Với chiến lược EBA, việc xem xét mỗi thuộc tính và loại bỏ các lựa chọn không được quan tâm sẽ được thực hiện cùng lúc. Với chiến lược hy sinh, mỗi thuộc tính sẽ được xem xét và chuyển sang thuộc tính tiếp theo nếu như lựa chọn chứa một giá trịn thuộc tính không được quan tâm.

  Trong ví dụ ma trận quyết định ở Bảng 7.1. Giả sử thuộc tính quan trọng nhất là giá. Giá trị ngưỡng sẽ được định nghĩa cho thuộc tính này, ví dụ là 60. Kết quả có được là các nhà cung câp số 7, số 8, số 9 và số 12. Bước tiếp theo là chọn thuộc tính quan trọng thứ hai, ví dụ như chất lượng gỗ. Giả sử chất lượng tối thiểu mong muốn là 3 sao. Giá trị ngưỡng này khi được áp dụng sẽ cho kết quả là nhà cung cấp số 8.

Như đã thấy, một số chiến lược ra quyết định không xem xét tất cả các giá trị thuộc tính của một lựa chọn. Có nghĩa là, chiến lược không xem xét sự đánh đổi của một giá trị của thuộc tính này với giá trị của thuộc tính khác. Các chiến lược này gọi là các chiến lược không bù (non-compensatory). Các chiến lược EBA, LEX và SAT là các chiến lược không bù.

Các chiến lược bù có thể được áp dụng nếu khi xem xét một thuộc tính của một lựa chọn với giả thiết rằng nó có thể được bù bởi một thuộc tính khác. Nói một cách khác, nếu xem xét hai thuộc ính của một lựa chọn, một giá trị thuộc tính hấp dẫn của một thuộc tính có thể bù cho một

giá trị không hấp dẫn của một thuộc tính khác. Các chiến lược WADD, EQQ, ADIFF, MCD và FRQ là các chiến lược có bù.

Khi phải đối mặt với số lượng lớn các lựa chọn. Thường bắt đầu với các lựa chọn không bù để đơn giản hóa hay rút gọn số các lựa chọn về một tập hợp lý. Khi tập các lựa chọn đủ nhỏ, các chiến lược bù có thể được áp dụng để xem xét các lựa chọn một cách chi tiết hơn. Tập các lựa chọn này gọi là tập xem xét (consideration set). Tập xem xét thường không lớn và có kích cỡ 5 hoặc 6.

Các chiến lược ra quyết định có thể được phân rã thành chuỗi các bước nhỏ: xem xét một giá trị thuộc tính, so sánh giá trị này với giá trị thuộc tính khác, lưu giá trị, và tiếp tục. Các bước này có thể được khái niệm hóa thành các khối xây dựng từ đố các chiến lược ra quyết định mức cao có thể được xây dựng. Các khối xây dựng đó gọi là các đơn vị xử lý thông tin cơ sở (Elementary Information Processing – EIP). Nếu chúng ta tính tổng cá EIPs cho từng chiến lược ra quyết định sẽ thấy một số chiến lược cần ít hơn nhiều EIPs so với các chiến lược khác.

Số các EIP là một phép đo nỗ lực nhận thức do đó có thể nói rằng một số chiến lược ra quyết đinh đòi hỏi it nỗ lực hơn các chiến lược khác. Có thể nói rằng, các chiến lược ra quyết định không bù đòi hỏi ít nỗ lực hơn các chiến lược ra quyết định có bù.

Các chiến lược ra quyết định cũng khác nhau ở mức độ chính xác. Độ chính xác thường được đo như lượng thông tin được xử lý trong một quyết định. Ví dụ, chiến lược EBA chỉ xử lý một giá trị thuộc tính trong khi WADD xử lý tất cả các giá trị thuộc tính. Chiến lược SAT không phải là chiến lược chính xác nhất do nó loại bỏ bất cứ lựa chọn nào tốt hơn lựa chọn đã được chọn. Theo kinh nghiệm chung, các chiến lược ra quyết định có bù chính xác hơn là các chiến lược ra quyết định không bù.

Các chiến lược với độ chính xác cao cần nhiều nỗ lực xử lý thống tin. Chọn một chiến lược ra quyết định do vậy là kết quả của sự đánh đổi giữa độ chính xác và nỗ lực cần thiết để đạt được độ chính xác. Lựa chọn chiến lược ra quyết định là một hàm của kỳ vọng của người dùng về nỗ lực bỏ ra và kỳ vọng của người dùng về độ chính xác đạt được.

7.6. CÁC KỸ THUẬT LỰA CHỌN PHƯƠNG ÁN

7.6.1. Shortlisting – Tạo danh sách ngắn

Một kỹ thuật hỗ trợ các nhà quản lý ra quyết định lựa chọn là kỹ thuật tạo danh sách ngắn. Các hệ thống cài đặt kỹ thuật này sẽ cho phép người dùng tạo một danh sách ngắn từ tập lớn hơn các lựa chọn sẵn có. Hệ thống sẽ cho phép người dùng hoán đổi qua lại giữa tập các lựa chọn và tập xem xét, thêm vào và loại bỏ các lựa chọn từ tập lựa chọn vào tập xem xét.

Kỹ thuật tạo danh sách ngắn có thể được hiện thực hóa theo các cách khác nhau như cho phép người dùng duy trì hai ma trận quyết định: một với tập lựa chọn và một với tập xem xét.

Một cách khác cài đặt kỹ thuật danh sách ngắn là cho phép người dùng đánh dấu một lựa chọn để xem xét sâu hơn. Trong trường hợp này, không cần duy trì một danh sách riêng nhưng các lựa chọn đã được gán nhã rõ ràng như là một phần của danh sách riêng rẽ. Các kỹ thuật cô lập

có thể được sử dụng để làm nỗi bật các lựa chọn được gán nhãn từ các lựa chón khác ví dụ như các lựa chọn trong danh sách ngwnx có thể có mầu nền khác v.v

Khi các lựa chọn được đánh dấu để xem xét, có thể sắp xếp thứ tự tập lựa chọn theo trạng thái đánh dấu sao cho các lựa chọn được đánh dấu được hiển thị ở trên cùng. Điều này rất có ích trog việc hỗ trợ ứng dụng ra quyết định dựa trên thuộc tính. Kết hợp kỹ thuật này với kỹ thuật xử lý sẽ cho phép người dùng sắp xếp thứ tự các lựa chọn được đánh dấu trong ma trận quyết định chính.

Một hệ thống thông tin áp dụng các kỹ thuật tạo danh sach ngắn sẽ hiển thị ma trân quyết định tới người dùng, tạo cơ hội cho người dùng xếp hạng các lựa chọn và tạo tập xem xét với giả thiết là người dùng có khả năng và sẵn sàng xem xét từng lựa chọn. Trong nhiều trường hợp, điều này không xảy ra. Đôi khi, người dùng không sẵn sàng xem xét lựa chọn một cách riêng rẽ. Ví dụ, khi người dùng bị áp lực về thời gian. Hoặc đôi khi, tập lựa chọn quá lớn để có thể đánh giá.

Để hỗ trợ tạo một danh sách ngắn tự động, lựa chọn có điều kiện có thể được sử dụng. Ý tưởng là đưa ra cho người dùng một số các mức ngưỡng để loại bỏ một số các lựa chọn từ ma trận quyết định. Hệ thống thông tin sẽ loại bỏ các lựa chọn mà người dùng cho rằng dưới hoặc trên các ngưỡng.

Ví dụ, với ví dụ bảng 7.1. Thay vì biểu diễn ma trận trực tiếp, một hệ thống thông tin có thể yêu cầu người dùng chọn một lựa chọn theo vùng, theo đánh giá chất lượng hoặc theo khoảng cách đến nhà máy và kho hàng.

Các lựa chọn có điều kiện hoạt động với giả thiết là người dùng có khả năng xác định các giá trị ngưỡng cho việc tạo danh sách ngắn. Một hệ thống thông tin có thể thông báo cho người dùng lựa chọn nào đáng xem xét.

7.6.2. Utility mapping – Tạo danh sách phương án mới theo ưu tiên của người dùng

Ngoài kỹ thuật tạo danh sách ngắn. Kỹ thuật ánh xạ tiện ích cũng được sử dụng để giúp các nhà quản lý trong việc chọn các lựa chọn. Ý tưởng là tạo một danh sách mới các lựa chọn theo thứ tự sở thích của người dùng. Danh sách được sắp xếp thứ tự này có thể có được bằng cách mô hình hóa sở thích người dùng và ghép cặp các sở thích này với các giá trị thuộc tính của các lựa chọn và tính toán một giá trị tiện ích tổng thể. Đây là điểm chỉ thị mức độ quan tâm của người dùng đối với lựa chọn. Một hệ thống có thể dùng giá trị tiện ích tổng thể để xếp hạng thứ tự các lựa chọn.

Việc tạo giá trị tiện ích tổng thể không phải là quá trình đơn giản. Quá trình này liên quan đến hai bước. Bước thứ nhất là định nghĩa hàm tiện ích cho từng thuộc tính để ánh xạc mỗi giá trị thuộc tính thành giá trị tiện ích. Bước này cần được thực hiện do các thuộc tính không thể so sánh trực tiếp với nhau. Bước thứ hai là cộng một trọng số vào mỗi thuộc tính. Điểm trung bình trọng số có thể được tính toán và xem xét cho tất cả các thuộc tính.

Áp dụng kỹ thuật này vào ví dụ với ma trận quyết đinh ở Bảng 7.1. Chúng ta sẽ định nghĩa một hàm tiện ích cho từng thuộc tính và xem xét các sở thích cho từng thuộc tính. Bảng 7.2 biểu diễn một tập các hàm tiện ích.

Bảng 7.2 Các ánh xạ tiện ích.

TT	Thuộc tính	Hàm tiện ích *U*
1	Vùng	*U1* = 0 nếu vùng A *U1* = 100 nếu vùng B
2	Chất lượng	*U2* = 0 nếu 2 sao *U2* = 50 nếu 3 sao *U2* = 100 nếu 4 sao
3	Khoảng cách	*U3, U4* = (KC lớn nhất – KC)/KC lớn nhất x 100
4	Giá	*U5* = (giá cao nhất – giá)/giá x 100

Chúng ta xem xét các hàm tiện ích trong ví dụ trên. Chú ý cách mỗi hàm tiện ích ánh xạ các giá trị thuộc tính vào dải giá trị tiện ích từ 0 đến 100. Các giá trị cận trên và cận dưới được chọn tùy tiện. Các loại hàm khác nhau cần cho các thang đo khác nhau. Để ánh xạ các giá trị dan định vào các giá trị tiện ích, hàm bước được sử dụng ở đó mỗi giá trị tương ứng với một bước. Để ánh xạ các biến thứ tự có thể dùng hàm bước hoặc hàm rời rạc. Cuối cùng các giá trị độ đo có thể sử dụng các hàm bước, hàm rời rặc hoặc hàm liên tục.

Bước thứ hai, giả sử khoảng cách và giá là các tiêu chí quan trọng nhất cho người dùng và các tiêu chí khác ít quan trọng hơn một chút. Điều này được phản ánh bởi đánh trọng số 20 phần trăm cho vùng, 20 phần trăm cho chất lượng gỗ, 13 phần trăm cho khoảng cách đến nhà máy và kho và 30 phần trăm cho giá. Do đó điểm cho mỗi lựa chọn được tính là:

$$U = 0.20 \times U1 + 0.20 \times U2 + 0.15 \times U3 + 0.15 \times U4 + 0.30 \times U5$$

Kết quả tính được ở là một bảng điểm như ở Bảng 7.3.

Bảng 7.3 Tính toán các giá trị tiện ích.

#	Vùng	Chất lượng	K. Cách N.Máy	K. Cách N.Kho	Giá	U
1	0	0	85	33	40	29.8
2	100	50	90	13	50	60.5
3	100	100	75	0	0	51.3
4	0	0	0	27	65	23.5
5	0	50	88	33	25	35.6
6	100	100	88	7	50	69.1
7	100	0	50	67	80	61.5
8	0	100	75	27	80	59.3
9	0	0	25	33	85	34.3
10	100	50	0	27	65	53.5
11	0	0	50	50	60	33.0
12	0	0	25	27	75	30.3

Bảng 7.4 Các nhà cung cấp được xếp hạng theo điểm giá trị tiện ích.

#	Vùng	Chất lượng	K. Cách N. Máy	K. Cách Kho	Giá
6	Region B	4 star	50	140	100
7	Region B	2 star	200	50	40
2	Region A	3 star	40	130	100
8	Region A	4 star	100	110	40
10	Region B	3 star	400	110	70
3	Region B	4 star	100	150	200
5	Region A	3 star	50	100	150
9	Region A	2 star	300	100	30
11	Region A	2 star	200	75	80
12	Region A	2 star	300	110	50
1	Region A	2 star	60	100	120
4	Region A	2 star	400	110	70

Sử dụng các giá trị tiện ích tình được trong Bảng 7.3, hệ thống có thể sinh ra một danh sách được sắp xếp theo thứ tự các nhà cung cấp như trong bảng 7.4 ở đó các lựa chọn tốt nhất đến tồi nhất được sắp xếp theo thứ tự từ trên xuống dưới.

Các kỹ thuật tạo danh sách ngắn và ánh xạ tiện ích có thể được kết hợp. Trong trường hợp đố, hệ thống có thể tạo một tập xem xét cho người dùng bằng cách thiết lập một giá trị tiện ích ngưỡng và đưa tất cả các lựa chọn vượt ngưỡng vào tập xem xét.

TÀI LIỆU THAM KHẢO

1. Hans van der Heijden, *"Designing Management Information Systems"*, Oxford University Press, 2009.
2. Kenneth C. Laudon and Jane P. Laudon, *"Management Information Systems"*, 12th Edition, Prentice Hall, 2012.
3. Bài giảng *"Cơ sở dữ liệu*", Học viện Công nghệ Bưu chính viễn thông, 2010.